

L6: ALTIUM SCHEMATIC CAPTURE II: BEST PRACTICES IN SCHEMATICS

ESE516: IoT Edge Computing

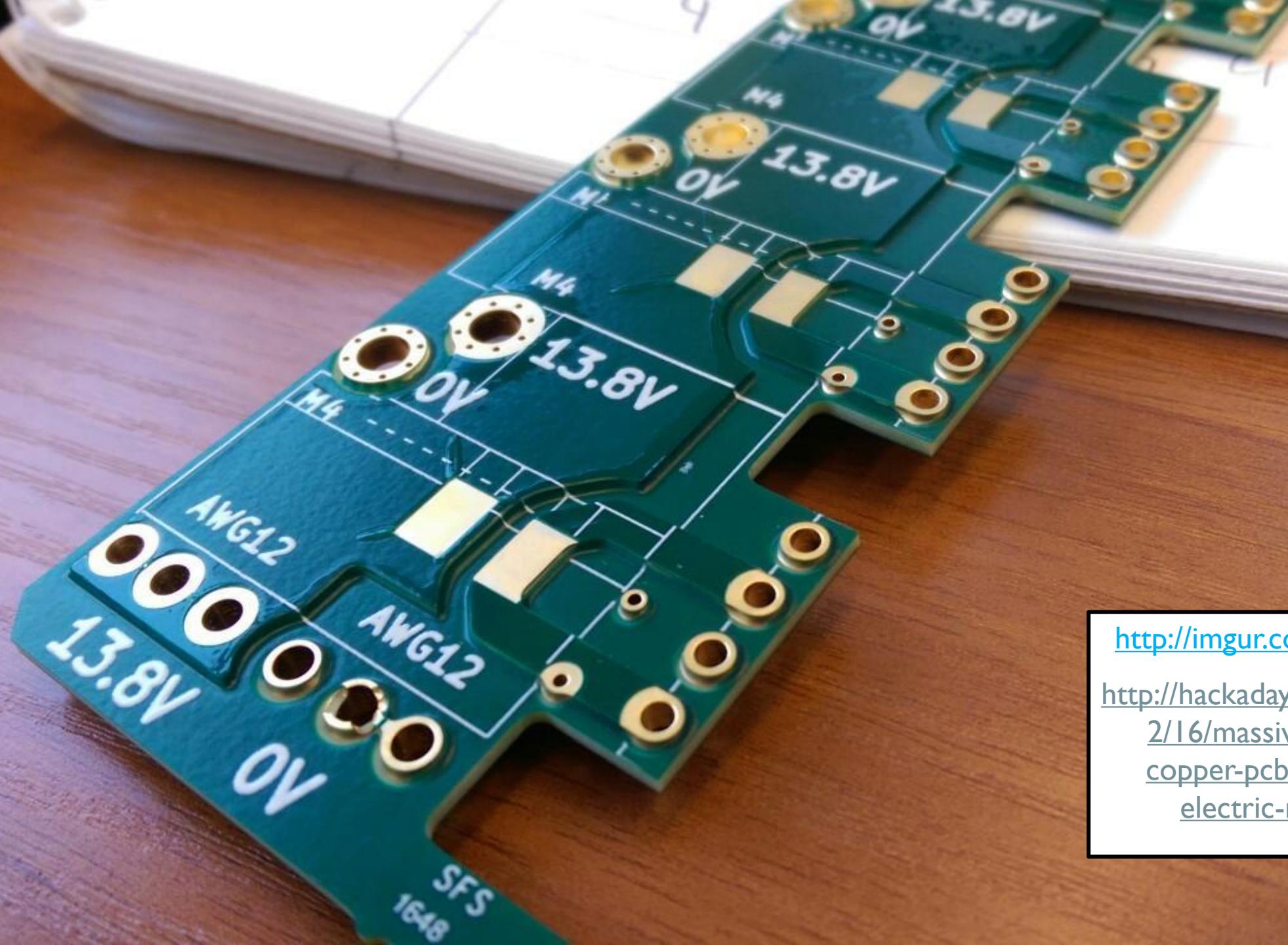
Monday, February 11, 2019

Eduardo Garcia – edgarc@seas.upenn.edu

20OZ COPPER PCB

<http://imgur.com/a/s2zcN>

<http://hackaday.com/2016/12/16/massive-20-oz-copper-pcb-enables-electric-racing/>



<http://imgur.com/a/s2zcN>
<http://hackaday.com/2016/12/16/massive-20-oz-copper-pcb-enables-electric-racing/>

TODAY'S LECTURE

LECTURE GOALS

1. Project Properties
2. Schematics Best Practices
3. Schematics Tips & Tricks

Altium
Designer.[®]

REVIEW

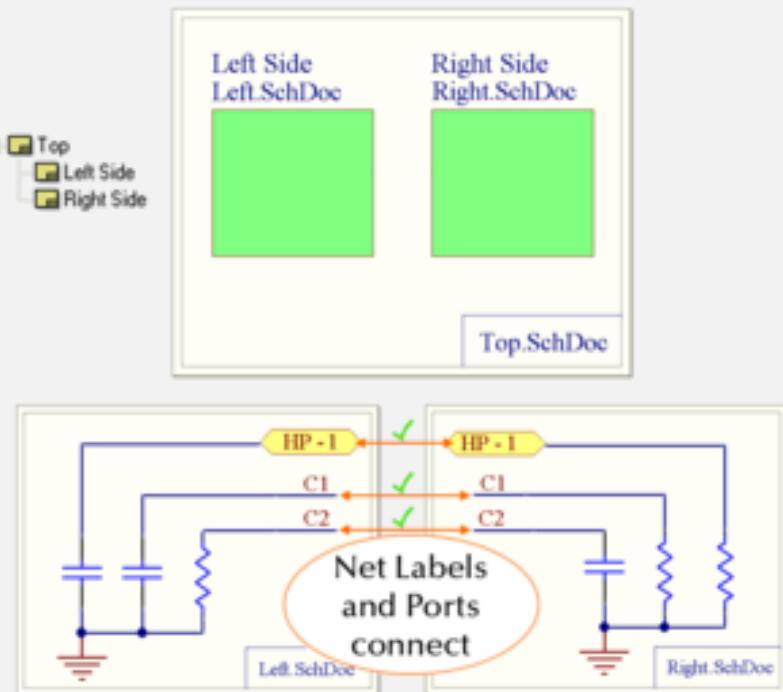
ALTIUM PROJECT

- The first step to success (in Altium) is to set the Altium project correctly to suit your needs.
- As we saw in the previous class (Schematic Tutorial) Altium has different ways in which it treats ports, nets and power ports

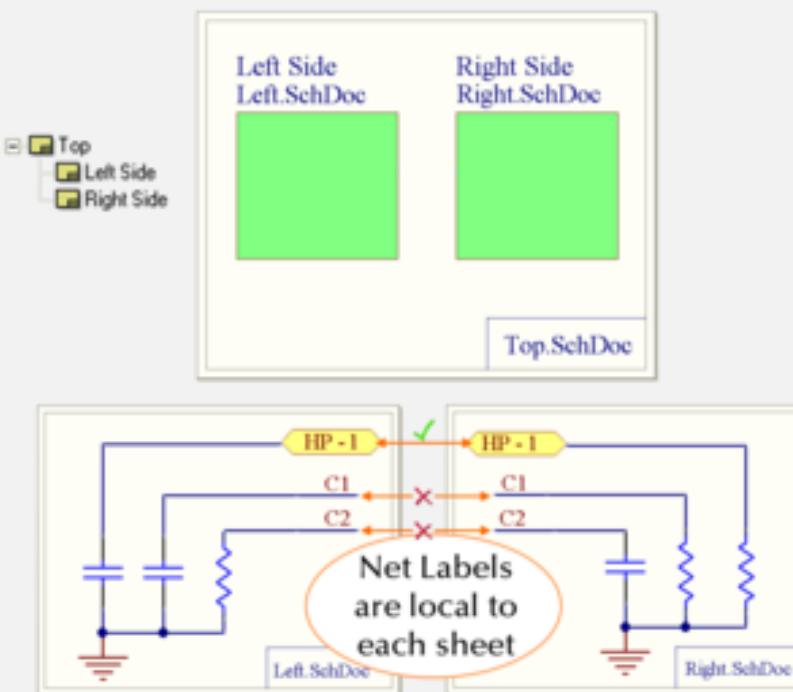
***Altium
Designer.***[®]

ALTIUM PROJECT

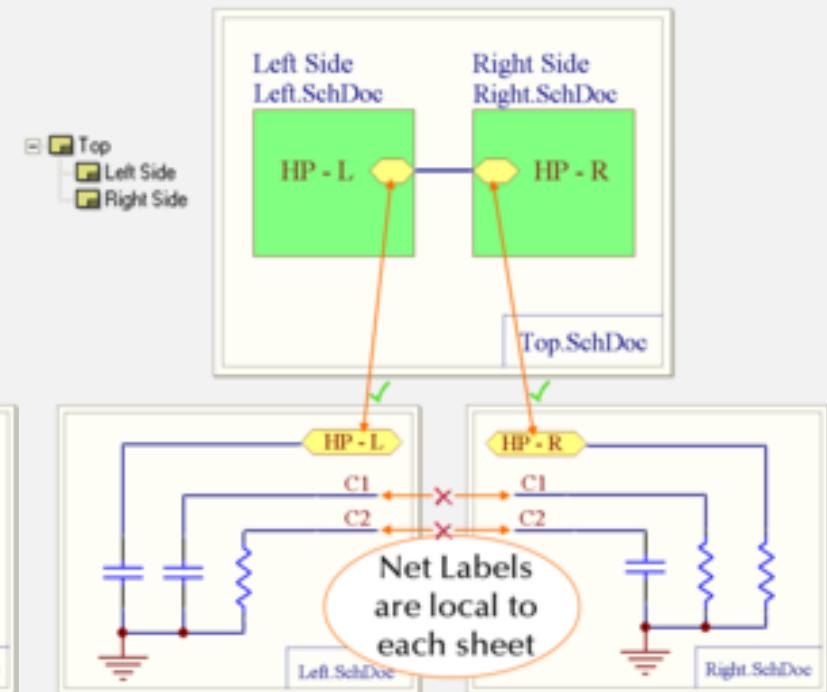
Global Project (Ports, Nets and Power Ports all global)



Flat: Ports and Power Ports are global, Net labels local



Hierarchical: Power Ports are Global, Nets local, ports local but have external access



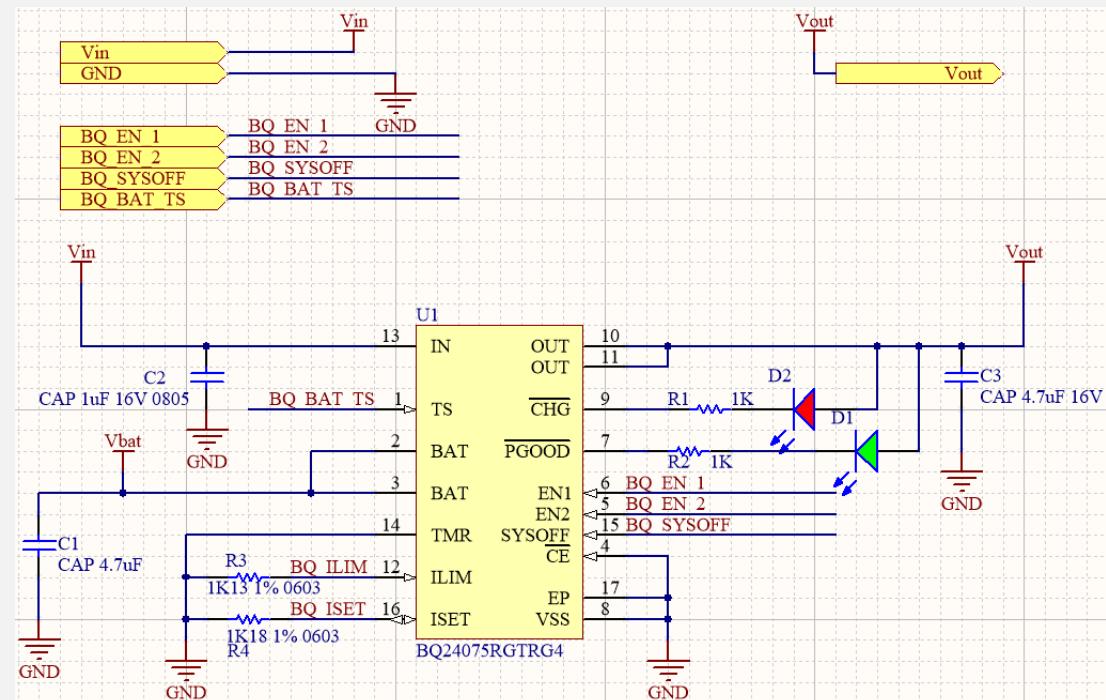
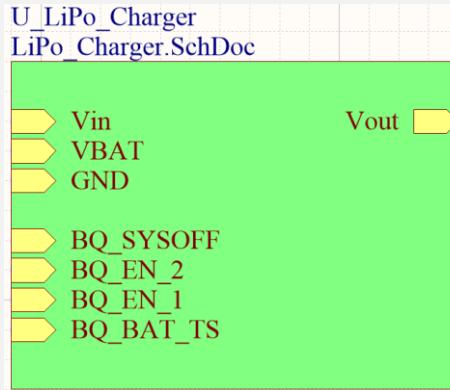
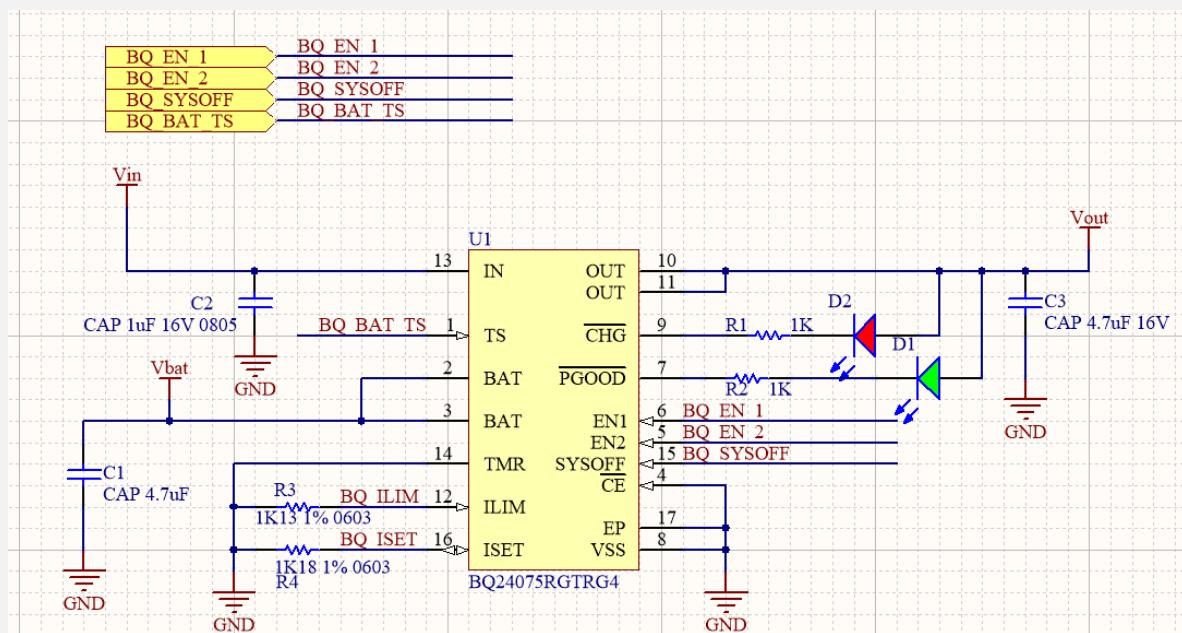
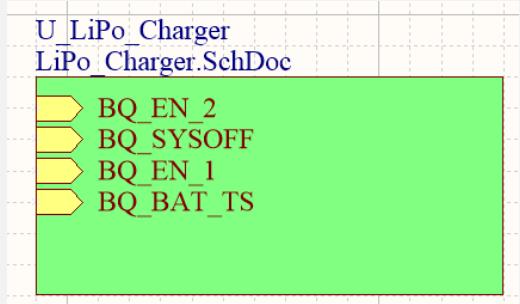
Used Last Class



There is another one that is more useful (in my opinion) on the next slide...

HIERARCHICAL VS STRICT HIERARCHICAL

Hierarchical (Power Ports
global, Net Labels and Ports
are local)



Another mode, called Strict Hierarchical, forces all ports, power ports, and net labels, to be strictly local.
Ports must be used to connect Power Ports between Sheets.

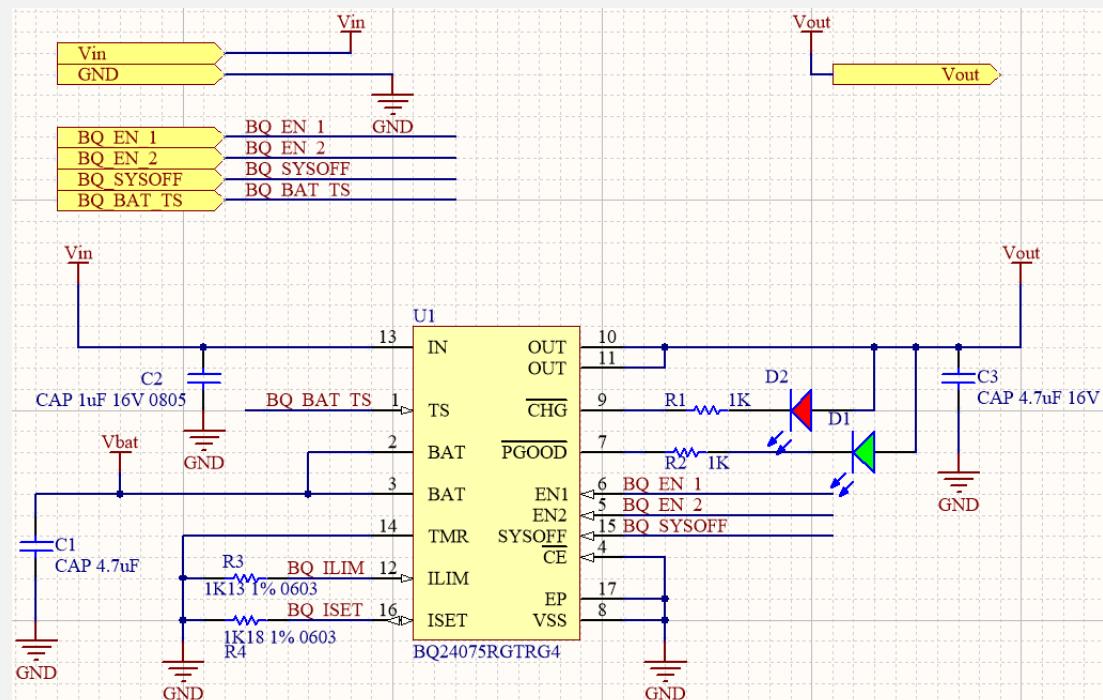
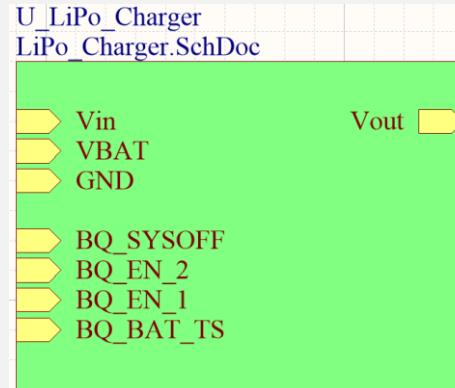
ALTIUM PROJECT (RECOMMENDED FOR THIS CLASS)

Strict hierarchical has the following advantages:

- Makes it very clear where power ports are coming from
- Complete visibility on signal that go in and out signal of a sheet symbol

Disadvantages:

- More wires (but no big deal!)



Another mode, called Strict Hierarchical, forces all ports, power ports, and net labels, to be strictly local. Ports must be used to connect Power Ports between Sheets.

NET NAMES ON A PROJECT

From:

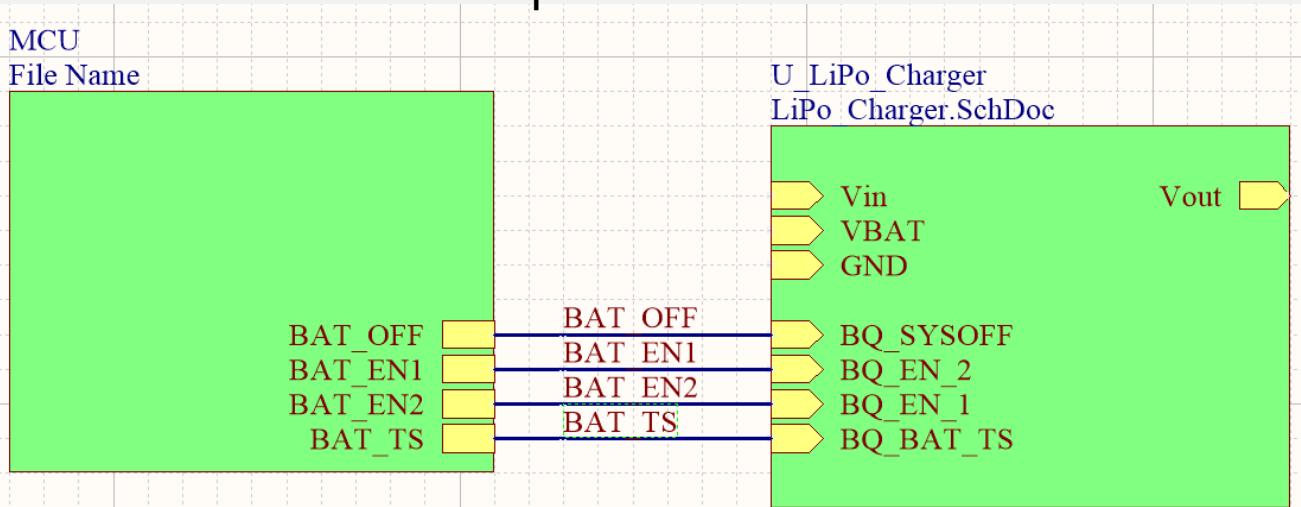
[https://www.altium.com/documentation/15.1/display/ADES/\(\(Creating+Connectivity\)\)_AD#!CreatingConnectivity-FlatDesign](https://www.altium.com/documentation/15.1/display/ADES/((Creating+Connectivity))_AD#!CreatingConnectivity-FlatDesign)

You cannot have multiple Net Labels with different names on the same net within a schematic sheet, this situation will be detected and flagged as an error when the project is compiled. However, it is legitimate to have multiple net identifiers on a net on different sheets that the net appears on.

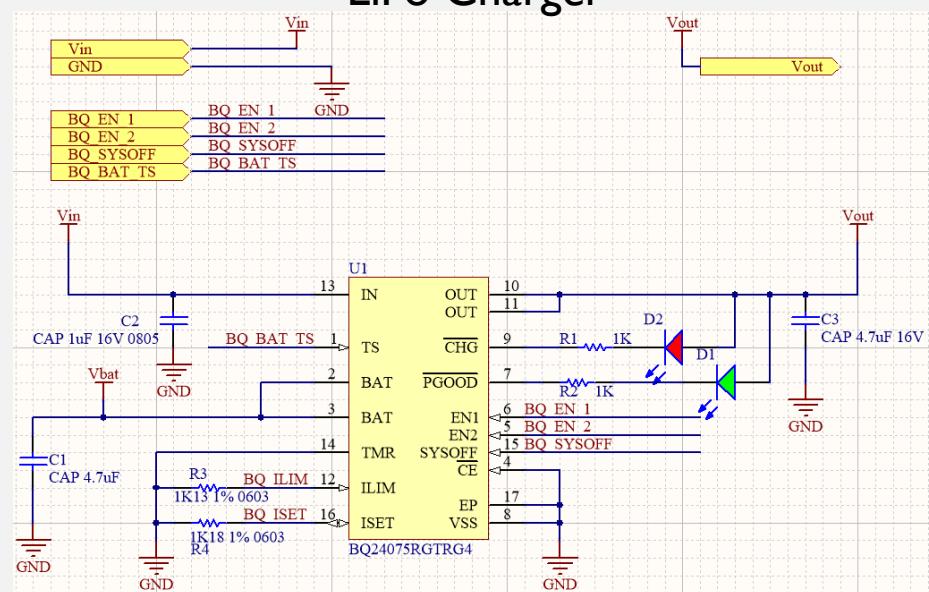
This ability allows the designer to:

- Change the name of a net at different levels in the hierarchy to better reflect its function on that sheet.
- Reuse a child schematic sheet without needing to rename nets on it.

Top Level



LiPo Charger

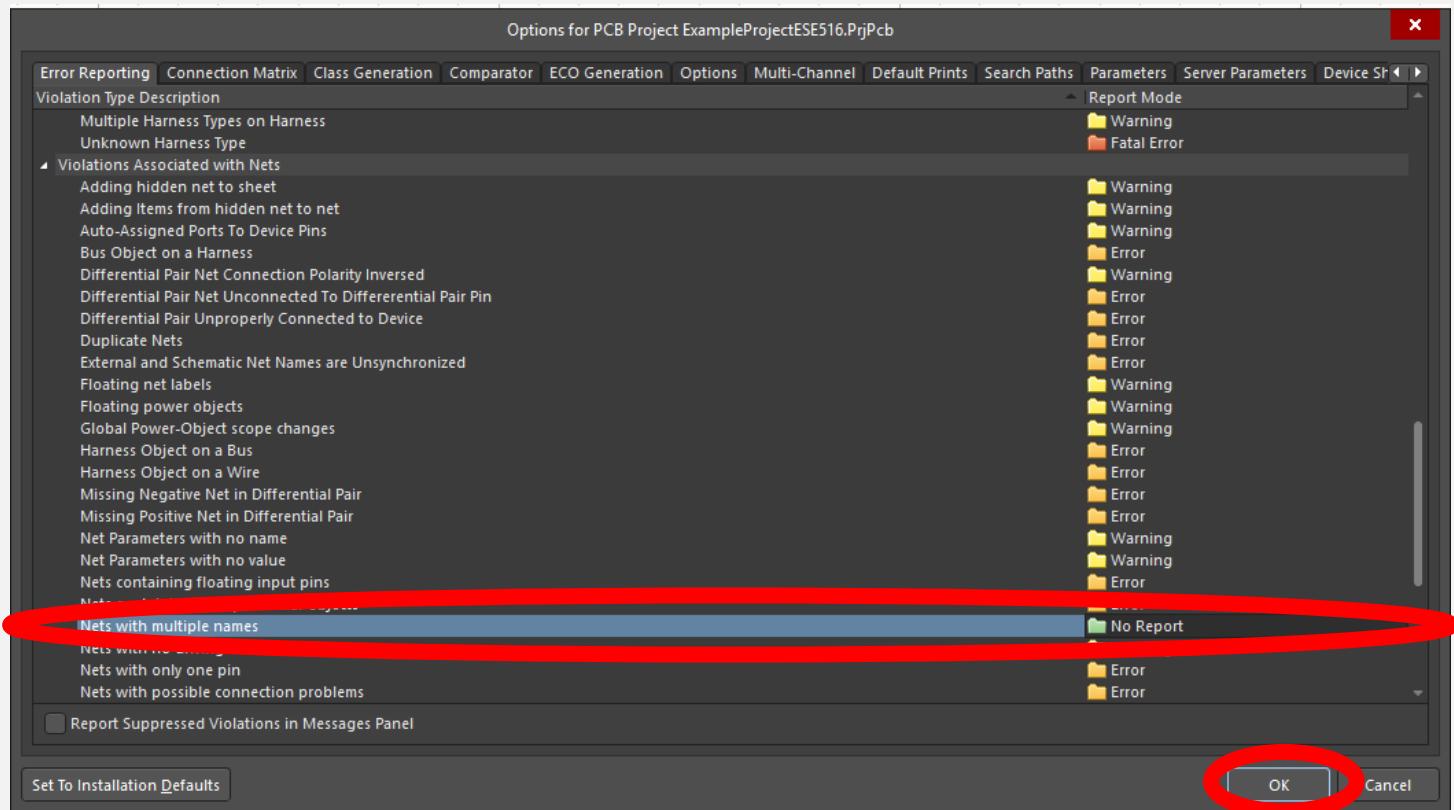


NET NAMES ON A PROJECT

This is a very common design style, but Altium's default behavior is to flag it as an error/warning!

You can edit the warning Altium gives you on this warning (and other) on “Project Options -> Error Reporting (Tab) ->Nets with Multiple Names”

I recommend changing this to “No Report”



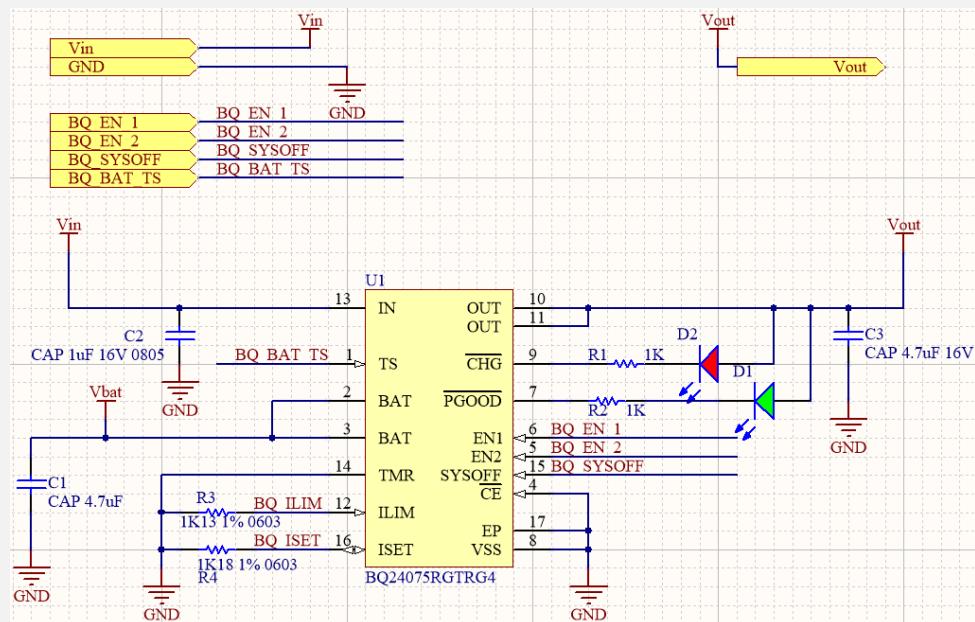
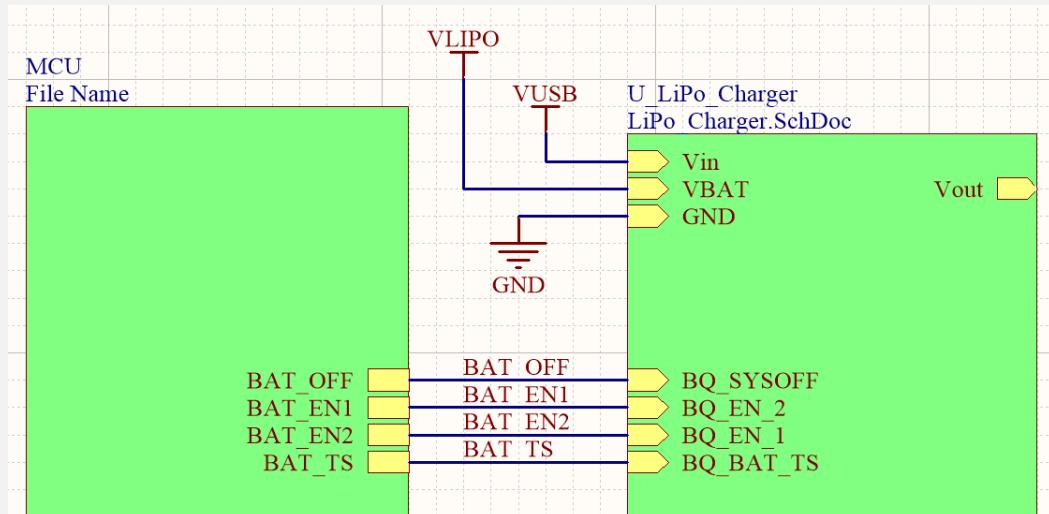
NET NAMES ON A PROJECT

With this style of net naming, we also have to tell Altium which name to prefer.

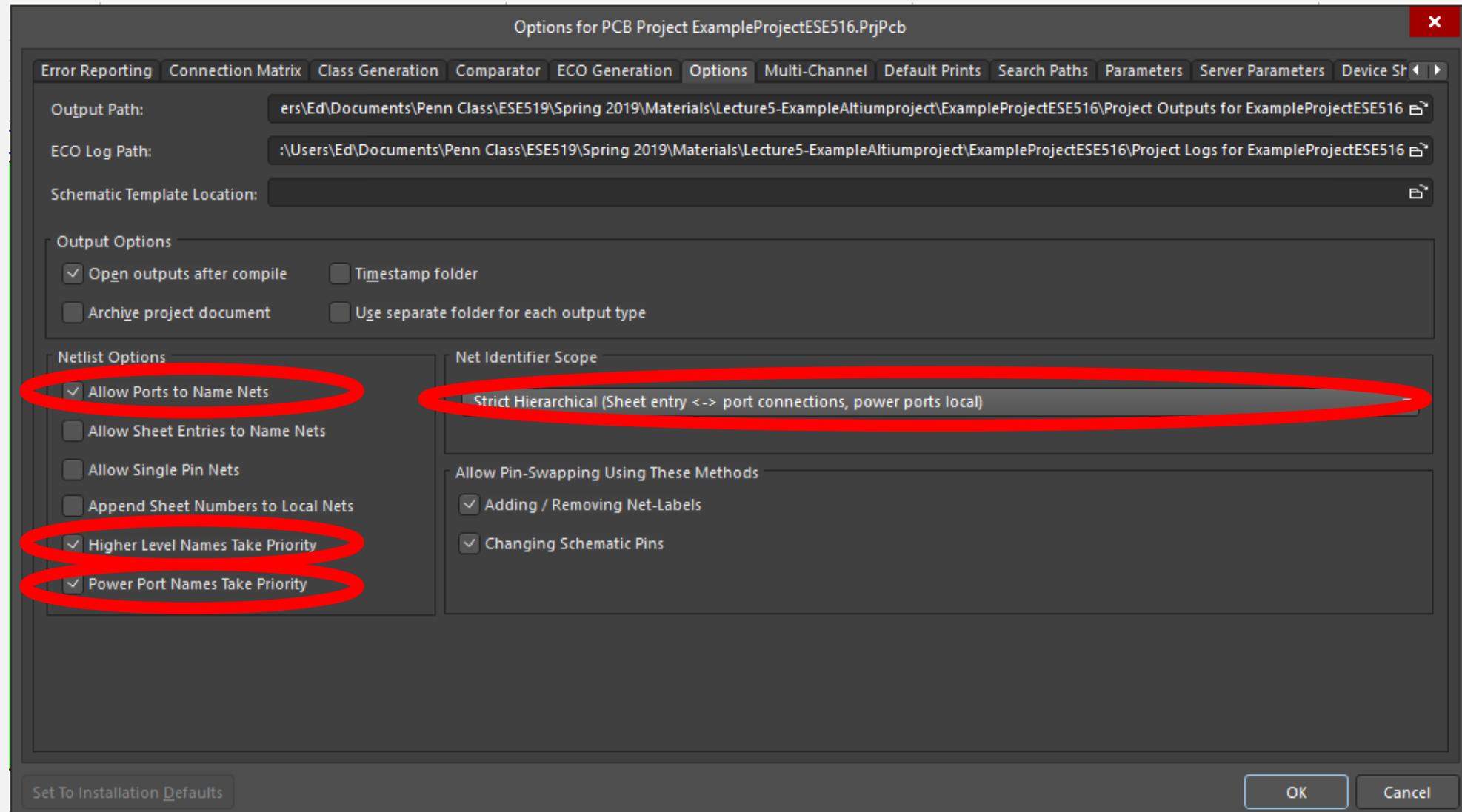
Nets can only have one name when we make the PCB. How do we deal with the fact that we allow multiple names on our Schematic Sheets?

Answer: It is good practice to tell Altium to prefer the net name from the highest schematic sheet hierarchy.

Usually the net name on the sheet of highest hierarchy that has the net is the most descriptive



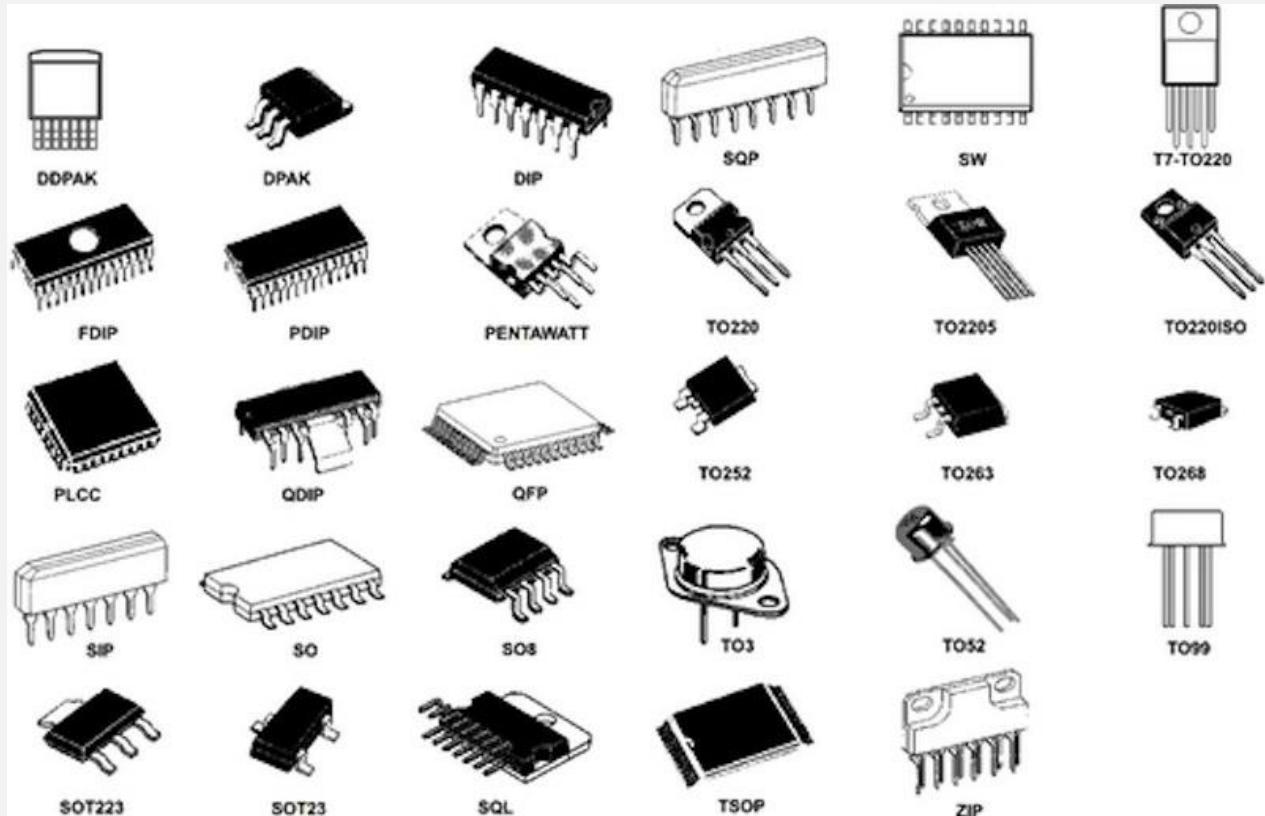
PROJECT OPTIONS TO CHANGE TO FOLLOW THESE RECOMMENDATIONS



COMPONENTS PACKAGES

COMPONENTS PACKAGES

- Component packages refer to the form factor a component has.
- Usually there are some standard in the industry, but certain manufacturers can have their proprietary component packages
- Packages may determine the following:
 - How easy it is to find a footprint compatible replacement
 - Diversity of component in same footprint (same device with different specifications)
 - How easy it is to manufacture
- List of most popular ones:
https://en.wikipedia.org/wiki/List_of_integrated_circuit_packaging_types



Our extensive package range provides maximum flexibility

	Ultra small	Very small	Small	Medium power
2 Pins	SOD962 (DSN0603-2) 0.6 x 0.3 x 0.3 WLCSP2 0.7 x 0.52 x 0.4 SOD882D (DFN1006-2) 1.0 x 0.6 x 0.37 SOD882 (DFN1006-2) 1.0 x 0.6 x 0.5 SOD523 1.2 x 0.8 x 0.6 SOD1608 (DFN1608D-2) 1.6 x 0.8 x 0.37	SOD323 1.7 x 1.25 x 0.95 SOD323F 1.7 x 1.25 x 0.7	SOD123F 2.6 x 1.6 x 1.1 SOD123W 2.6 x 1.7 x 1.0	SOD128 3.8 x 2.6 x 1.0
3 Pins	SOT883B (DFN1006B-3) 1.0 x 0.6 x 0.37 SOT883 (DFN1006-3) 1.0 x 0.6 x 0.5 SOT663 1.6 x 1.2 x 0.55 SOT416 1.6 x 0.8 x 0.77	SOT323 2.0 x 1.25 x 0.95 SOT1061 (DFN2020-3) 2.0 x 2.0 x 0.65	SOT23 2.9 x 1.3 x 1.0	SOT89 4.5 x 2.5 x 1.5
4/5 Pins	WLCSP4* 0.76 x 0.76 x 0.61 SOT1194 (DFN1010C-4) 1.0 x 1.0 x 0.55 WLCSP5* 1.51 x 1.14 x 0.65 SOT665 1.6 x 1.2 x 0.55	SOT353 2.0 x 1.25 x 0.95	SOT143B 2.9 x 1.3 x 1.0 SOT753 2.9 x 1.5 x 1.0	SOT223 6.5 x 3.5 x 1.65 Power-SO8 4.9 x 3.95 x 1.0
6/8 Pins	SOT1202 (DFN1010B-6) 1.0 x 1.0 x 0.35 SOT891 (DFN1010-6) 1.0 x 1.0 x 0.5 SOT886 (DFN1410-6) 1.45 x 1.0 x 0.5 SOT666 1.6 x 1.2 x 0.55 WLCSP6* 1.6 x 1.15 x 0.65	SOT363 2.0 x 1.25 x 0.95 SOT1220 (DFN2020MD-6) 2.0 x 2.0 x 0.65 SOT1118 (DFN2020-6) 2.0 x 2.0 x 0.65	SOT457 2.9 x 1.5 x 1.0 SOT505 3.0 x 3.0 x 1.1	SOT873-1 (DFN3333-8) 3.3 x 3.3 x 1.0 SOT96 4.9 x 3.9 x 1.75
> 8 Pins	WLCSP9* 1.16 x 1.16 x 0.61 SOT1157 (DFN1712-B) 1.7 x 1.2 x 0.5 SOT983 (DFN1714U-8) 1.7 x 1.35 x 0.5	WLCSP16* 1.9 x 1.97 x 0.65 SOT1178 (DFN1710-9) 2.1 x 1.0 x 0.5 WLCSP24* 2.45 x 2.4 x 0.65	SOT2116 2.5 x 1.0 x 0.5 SOT2118 2.5 x 1.2 x 0.5 SOT2119 2.5 x 1.2 x 0.5	SOT552 3.0 x 3.0 x 1.1 SOT617 (DFN5050-32) 5.0 x 5.0 x 1.0 SOT510 9.7 x 4.4 x 1.1

* The exact position of the balls and package dimensions vary.

Taken from
<http://dangerousprototypes.com/blog/2012/09/20/smd-package-cheat-sheets/>

KNOW YOUR PACKAGES...

<i>comparison</i>	Metric code	Imperial code	<i>comparison</i>
0.1x0.1 mm	0402	01005	0.01x0.01 in (10x10 mils)
	0603	0201	
	1005	0402	
	1608	0603	
1x1mm	2012	0805	0.1x0.1 in (100x100 mils)
	2520	1008	
	3216	1206	
	3225	1210	
	4516	1806	
	4532	1812	
	5025	2010	
	6332	2512	
	<i>Actual size</i>		0.5x0.5in (500x500 mils)

FOOTPRINT SELECTION

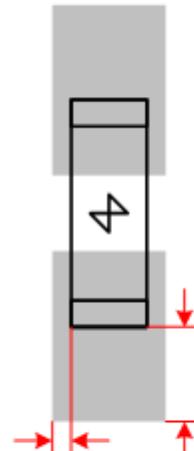
- Some components come with multiple footprints. This is true for all passives.
- Remember the IPC? They have footprints for low, middle, and high density.
 - **L (Least)** – tightest, for wearables
 - **M (Most)** – largest, for hand soldering
 - **N (Nominal)** – middle of the road
- We'll always be using **M (Most)** because we'll be hand placing and soldering everything in this course.

SMD RESISTOR FOOTPRINTS DEPENDING ON GOALS

Manual Placement
Manual Soldering

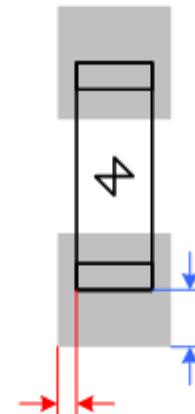
Manual Placement
Oven Soldering

Machine Placement
Oven Soldering



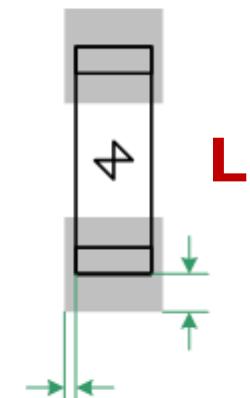
M

Footprint goal:
Extended areas for
placement uncertainty
and soldering tool contact



N

Footprint goal:
Less extended areas for
placement uncertainty
with higher density design



L

Footprint goal:
Optimal areas for the
highest fabrication-
possible density design

SIZING COMPONENTS

- Pay attention to component sizing – you can't always get what you want.
 - But you get what you **need**.
- Large capacitance, large power need for resistors, inductors
 - You cannot get small packages for these things
 - Or, if you can get them, could be exotic materials and expensive to use
 - Check DigiKey before building it into your design
- **For this project, assume 0805 components for everything, unless you need bigger.**
 - We're trying to standardize the components.
- Aim for ceramic capacitors – they're cheap, good, and non-polarized.
 - Tantalum are great, but expensive.
 - Electrolytic can be useful, especially for high capacity. They're polarized, though!



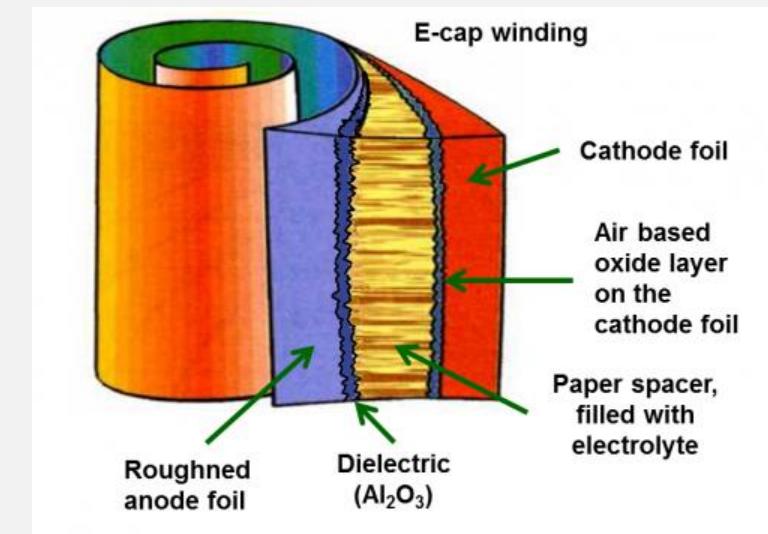
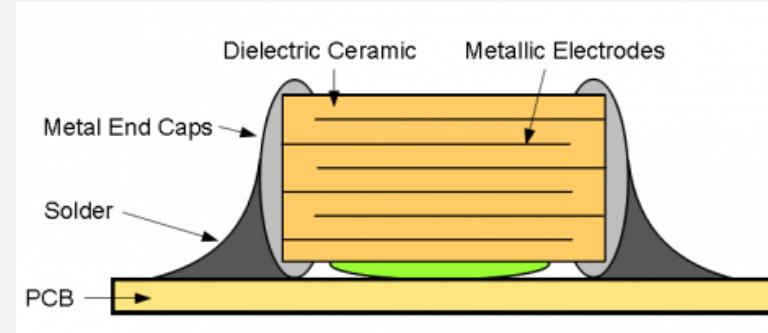
CAPACITOR SELECTION

- Capacitors can be some of the most complex parts to choose – crazy, right?
- Rule of thumb for ceramic caps: have a rating at least **twice the voltage** that you're expecting
- Many different chemistries, leakage current, storage density, temperature stability, long term stability... etc.
- For this class, we're requiring you to use only:
 - **Surface mount ceramic capacitors (MLCC) (Except for certain scenarios...)**
 - **0805 or 0603 imperial component packages (0805 preferred)**
- **Pick an appropriate voltage rating!**



DE-RATING CAPACITORS

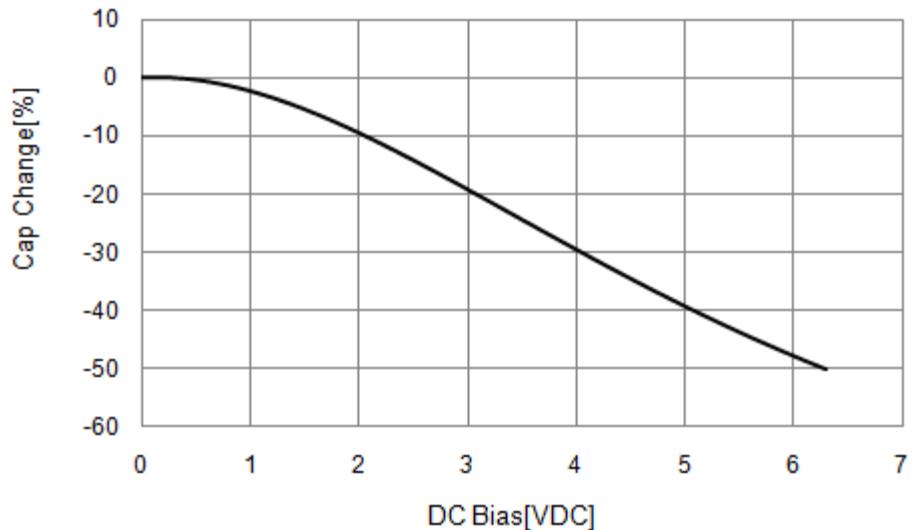
- De-rating is running below the rated voltage, current, or other value
- Why?
 - Running at the edge of the limits can shorten the lifetime of the part
 - Survive a wide temperature/pressure range while still retaining important characteristics (capacitance, ESR)
 - Some components lose their capacitance value as they approach the rated voltage
 - Running the component overvoltage can lead to catastrophic effects – tantalums especially
<https://youtu.be/XgKLlvEI4U?t=21s>
- Different effects on different capacitor construction
 - Ceramics rule of thumb: 50% derating on voltage
- General rule of thumb – use a capacitor rated for 2-3x the max voltage



MLCC CAPACITOR DERATING

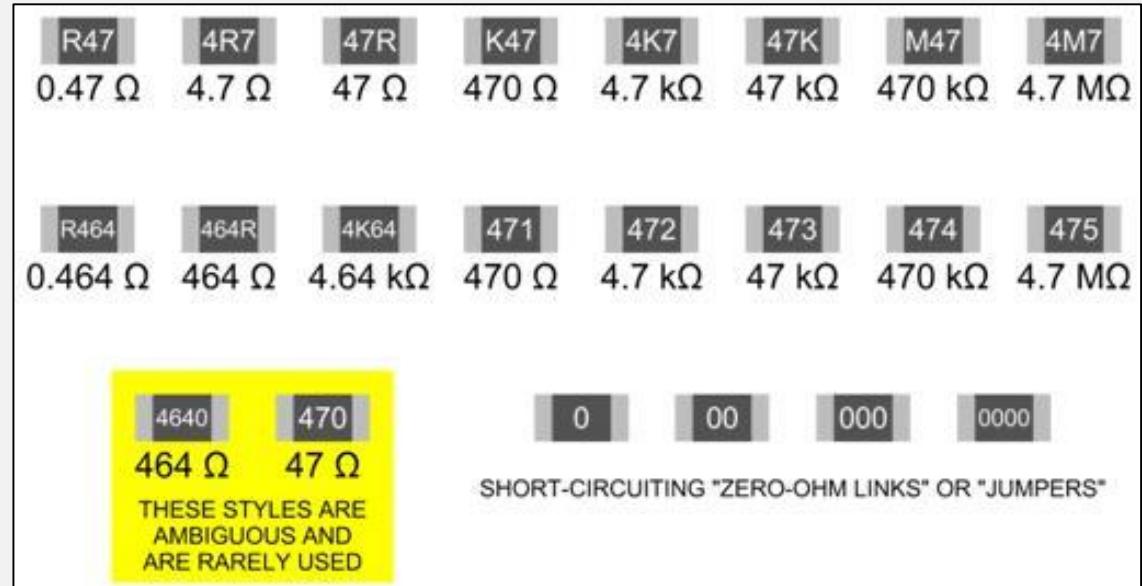
- MLCC experience an effect called “DC (direct current) bias characteristic”, also called Capacitor Derating. It is an effect of the technology
- See <https://www.murata.com/en-us/support/faqs/products/capacitor/mlcc/char/0005> for an analysis on why this happens
- Some other capacitors, like electrolytic or tantalum, do not have this effect.

Examples of direct current voltage characteristics



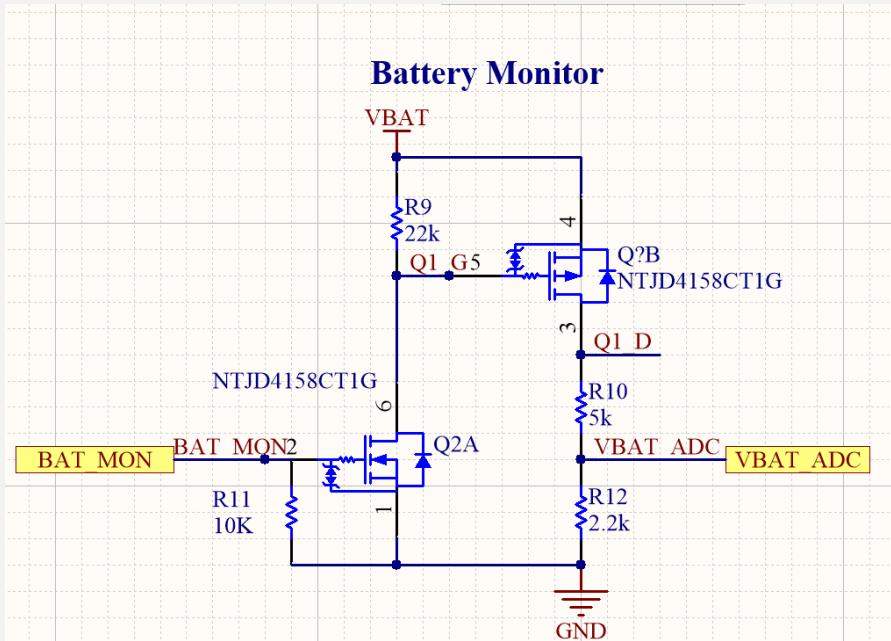
RESISTOR SELECTION

- Review the naming convention on SMT (surface mount) resistors to the right
- For this class, we're requiring you to use only:
 - **Surface Mount Chip Resistors**
 - **0805 or 0603 imperial component packages (0805 component packages preferred)**
- **Choose an appropriate power rating!**
 - $P = (I^2) * R = (V^2) / R = I * V$



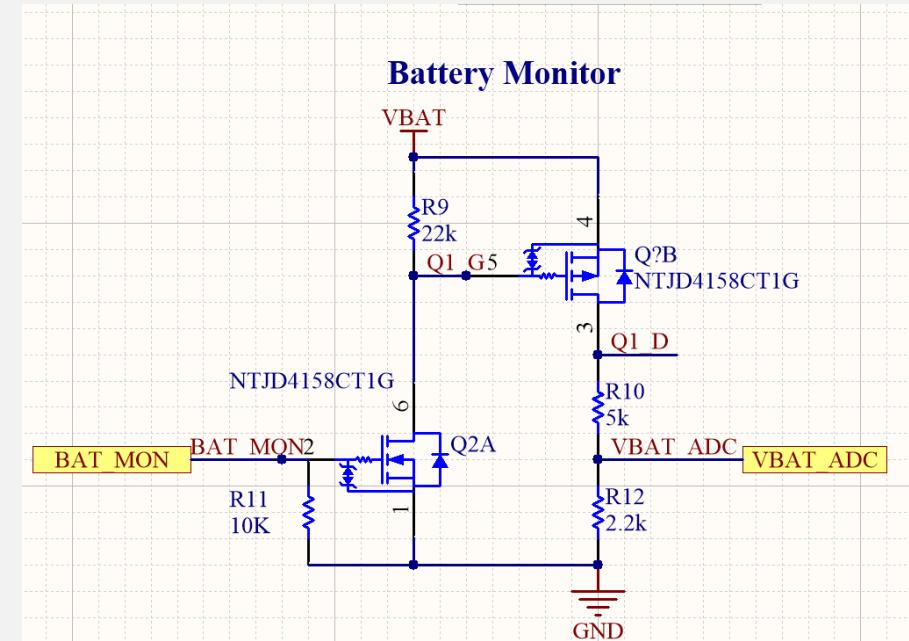
GET THE RESISTORS/CAPACITORS FOR EACH VALUE FROM THE VAULT!

DO



- Each individual resistor/capacitor is its own component obtained from the Altium Vault

DON'T

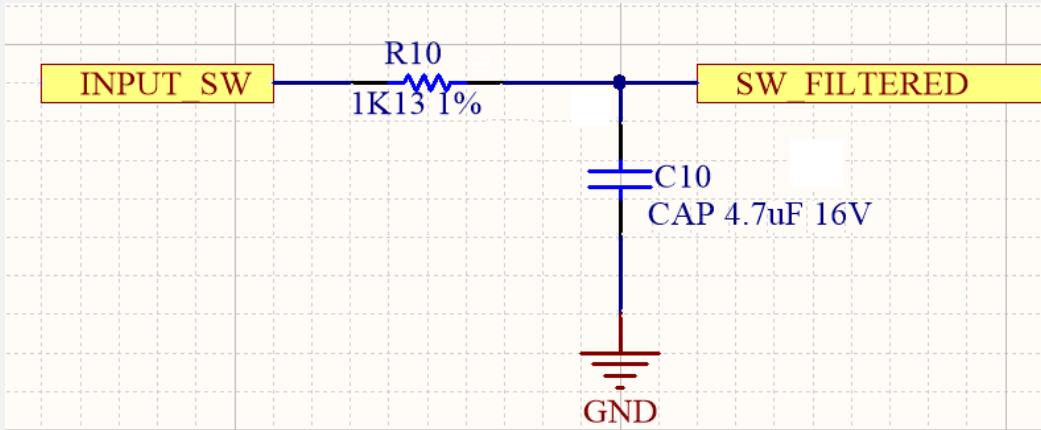


- The resistors/capacitors come from a single one that was copied-pasted and the values changed.
- Designing on borrowed time – will have to pay the consequences one day!!!

SCHEMATICS DRAWING BEST PRACTICES

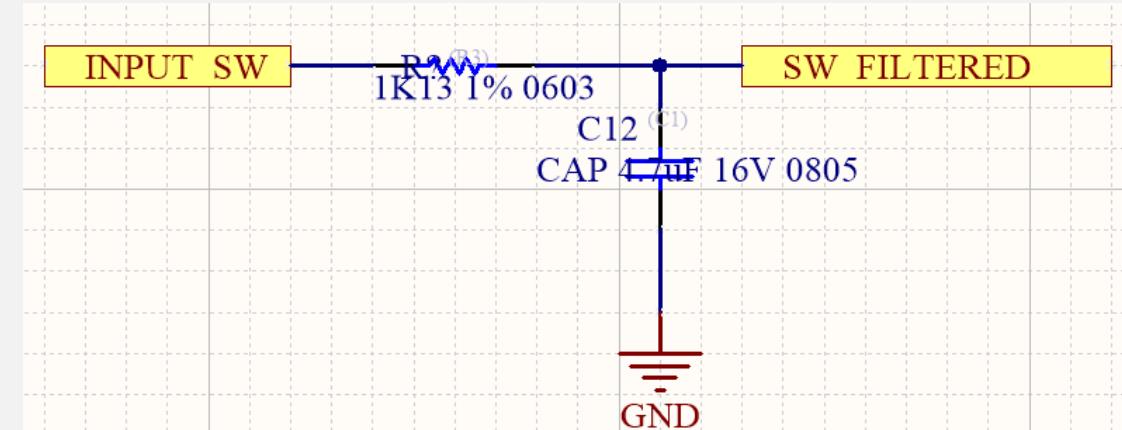
READABILITY

DO



- Readable
- Designators close to component
- Only information needed is presented

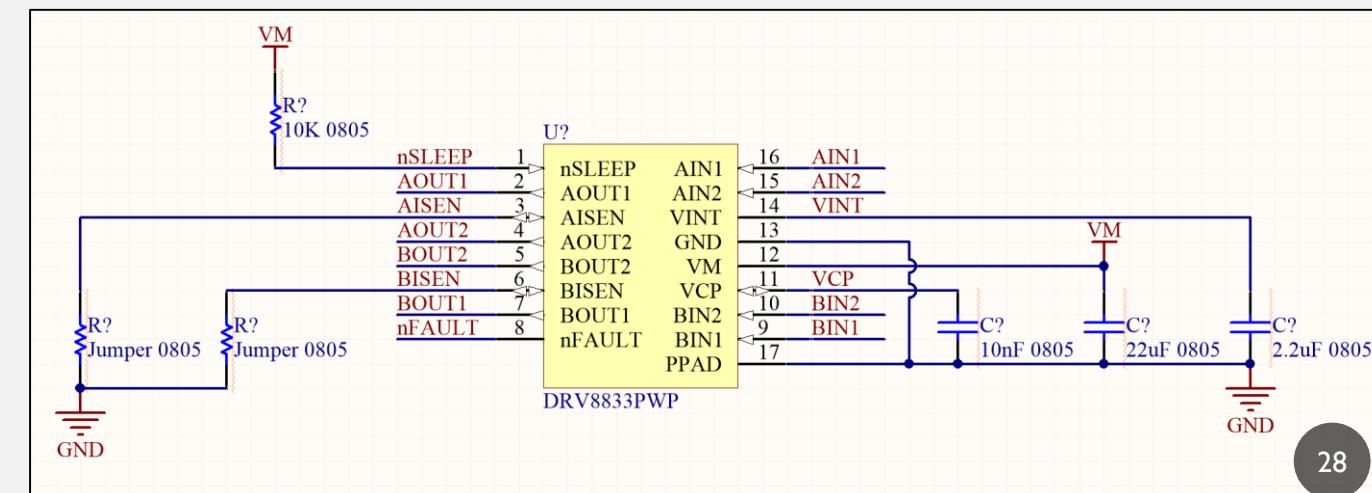
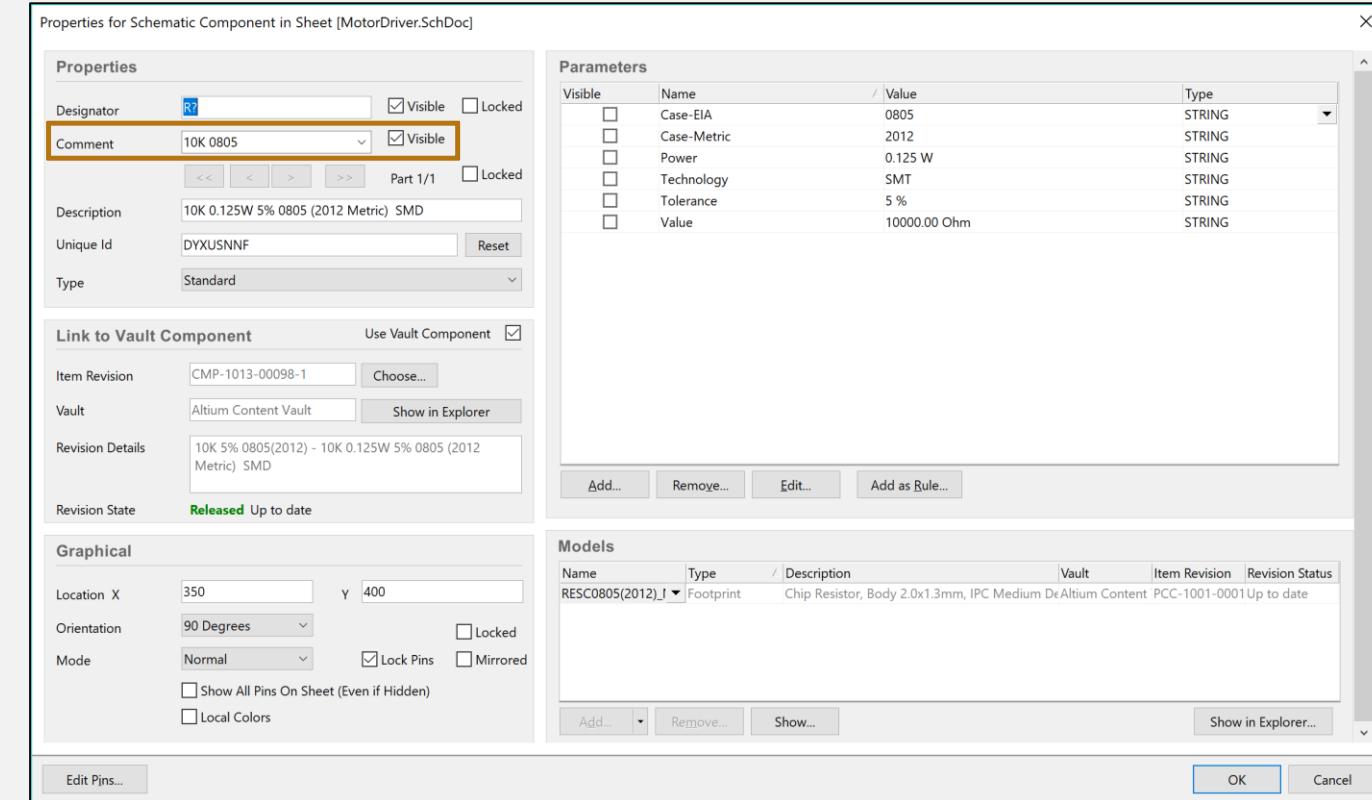
DON'T



- Not readable
- Designators and comments hit wires/hidden
- Extra information present (it does not matter that the capacitor is 0805 for the functioning of the circuit!). However you may leave it if it is important for your device/BOM.
- Schematics like this will be heavily penalized in this class!

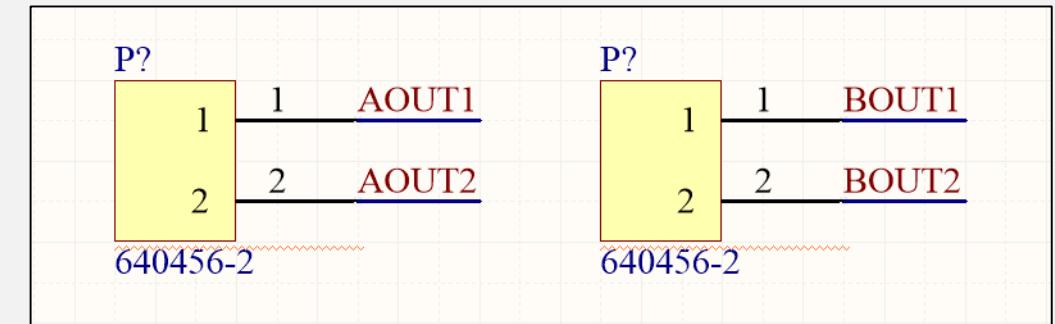
COMPONENT COMMENT INFORMATION

- Schematics should be clear – and components should have an appropriate level of information.
- For example, the **Comment** field for a component will be visible by default. I edit this to only communicate basic and critical information. Some scenarios include:
 - Specific tolerance
 - Specific power ratings
 - Specific package sizing
- For example, my schematic for the motor driver calls out the package size, as 0805 is the only package I will be using.

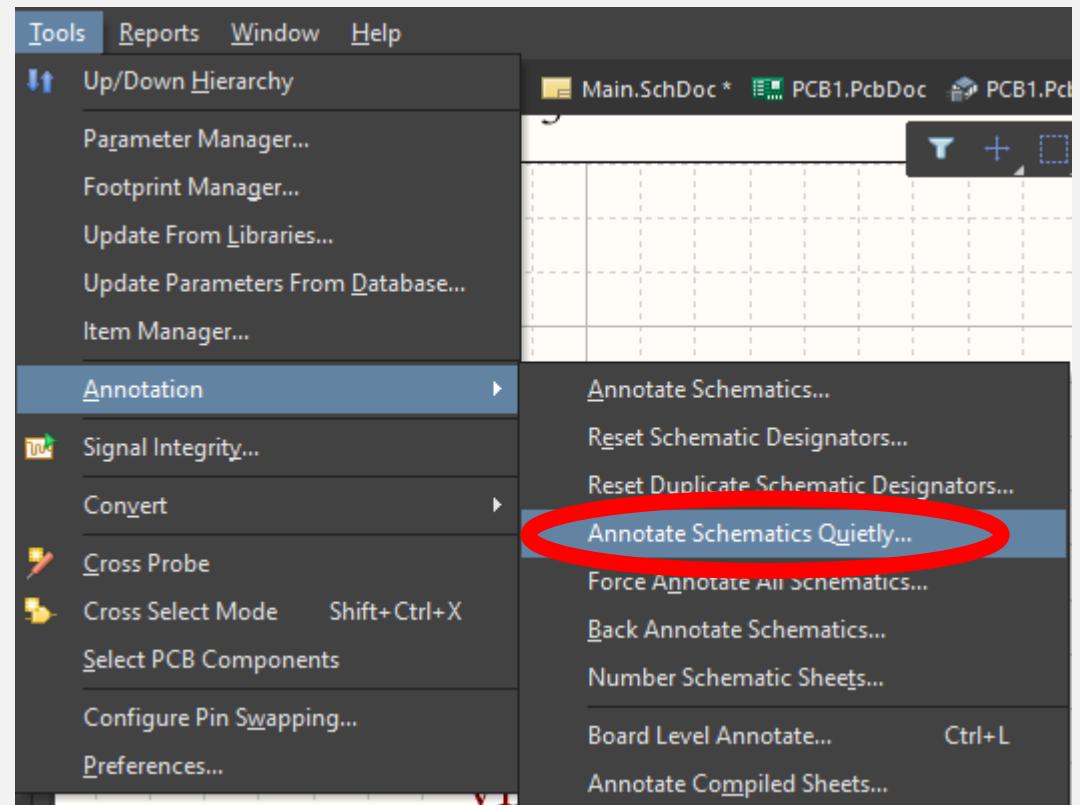


ASSIGN DESIGNATORS

- Components are known by their designators – such as C22, R30, or U3.
- In A2, you should have modified your default designator for your ICs for be **U?** – as this is an IC, and the question mark allows us to do auto-assigning.
- To solve this, you can manually name each component. Or, use the Annotation Tool to do it for you.
 - **Annotate Schematics...** goes into detail about how each component is labelled.
 - **Annotate Schematics Quietly...** doesn't bother you with the details. You'll probably use this the most.
 - The **Reset Schematic Designators** commands are good if you want to solve a duplicate name issue, or start the naming from scratch

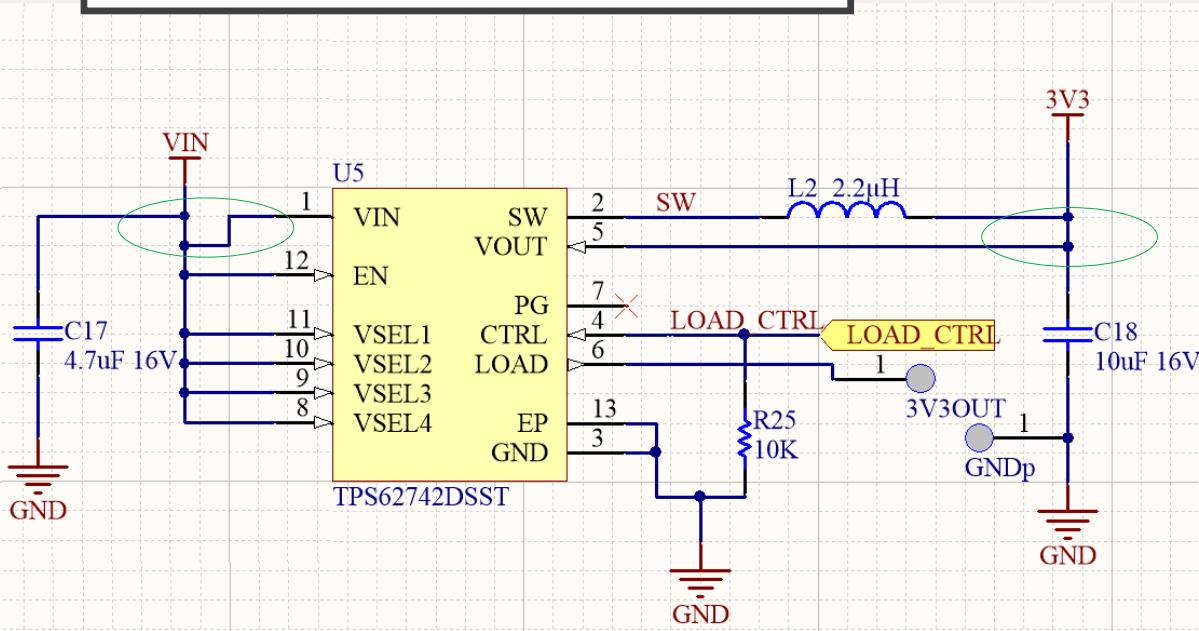


If you have multiple components with the same name, like **P?** and **P?** in the image above (or anything identical, C2 C2, etc), then you will have a compiling error. You cannot have two components named the same thing – the PCB would not be able to be assembled.

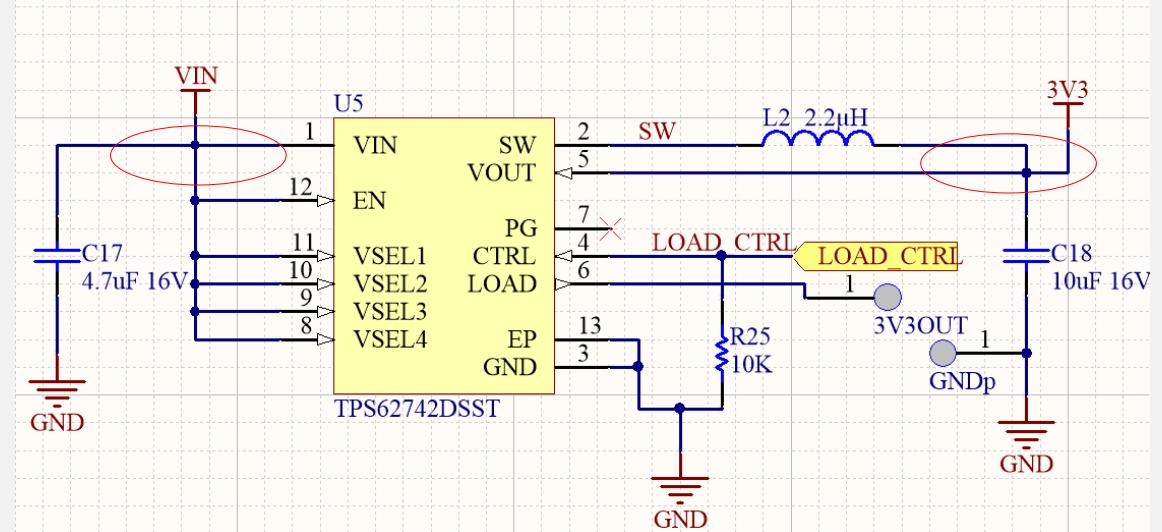


WIRING

DO



DON'T

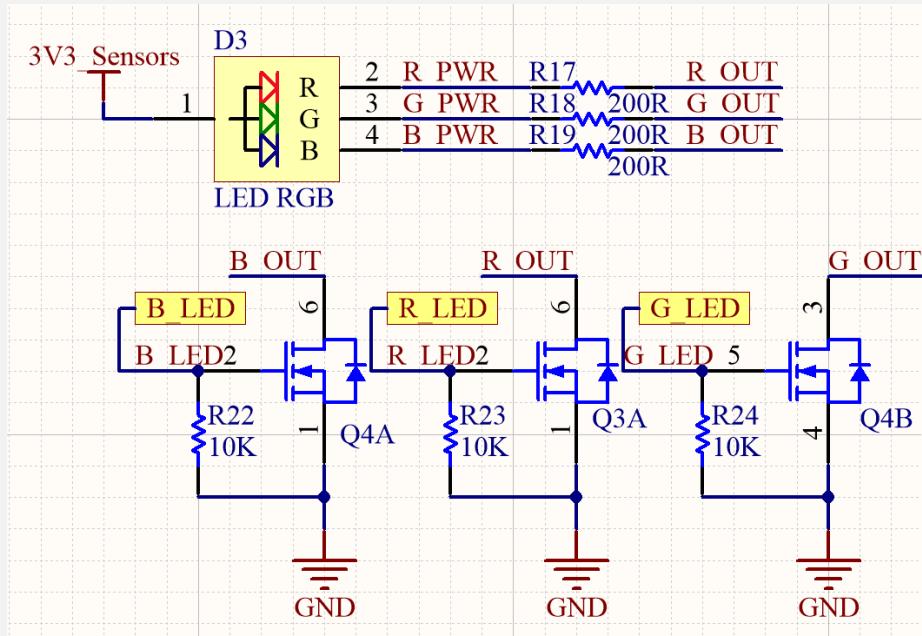


- Wire intersection only to three wires max.
- Use net names to make it clearer

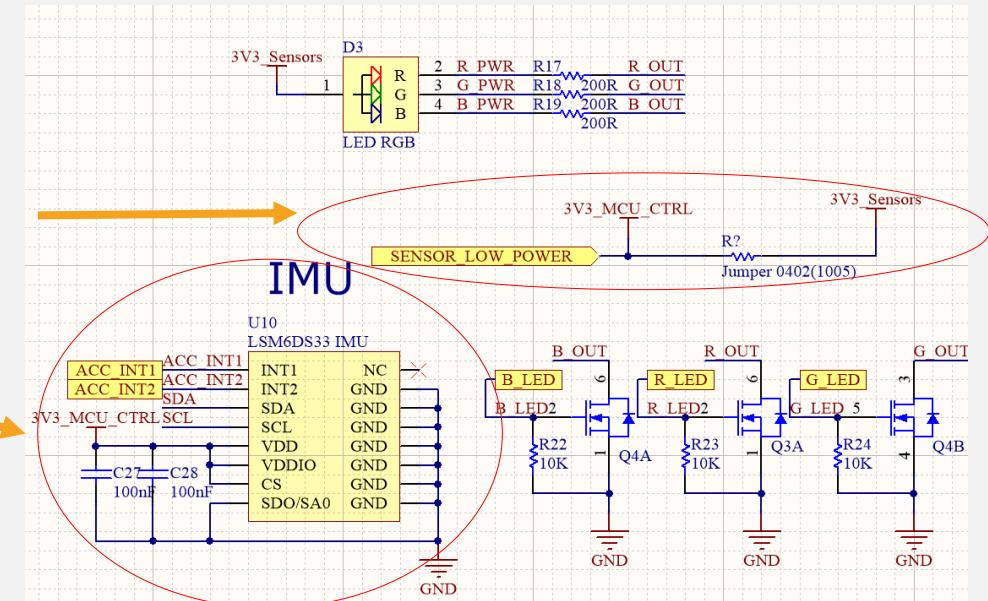
- Four wire intersection are frowned upon (it can be very easy to confuse with two wires just overlapping!)
- Not using net names to our advantage

USING NETS AND KEEPING CIRCUITS TOGETHER

DO



DON'T



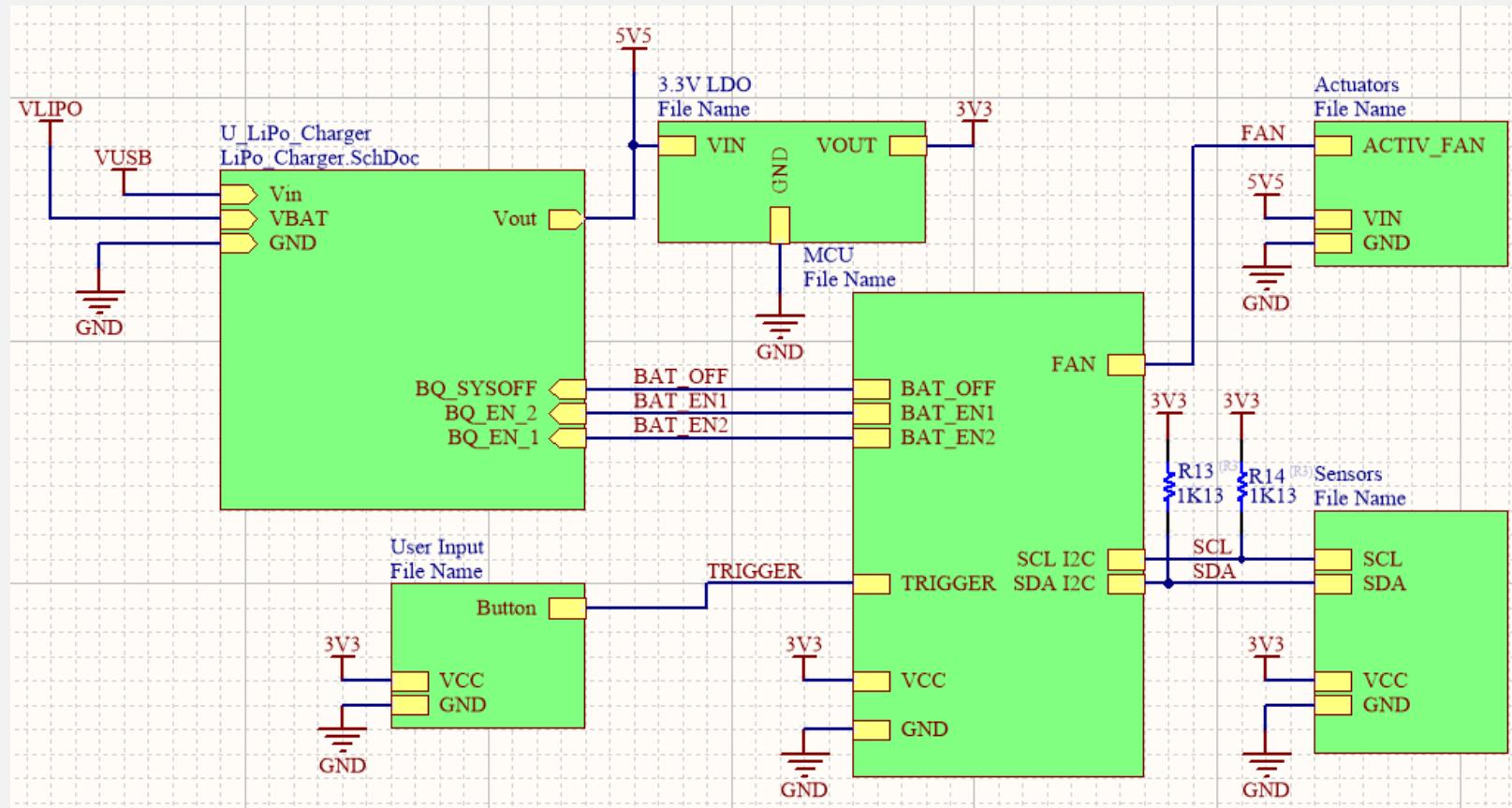
Unrelated Circuit
disrupts reading
flow

- Keep circuits together and use Nets to improve readability (in this example, wiring directly could have been good too)

- Do not use nets to divide circuits and then separate them in a way that makes analyzing them harder for the reader!

CONNECTIONS AND NETS ON TOP SCHEMATIC

DO

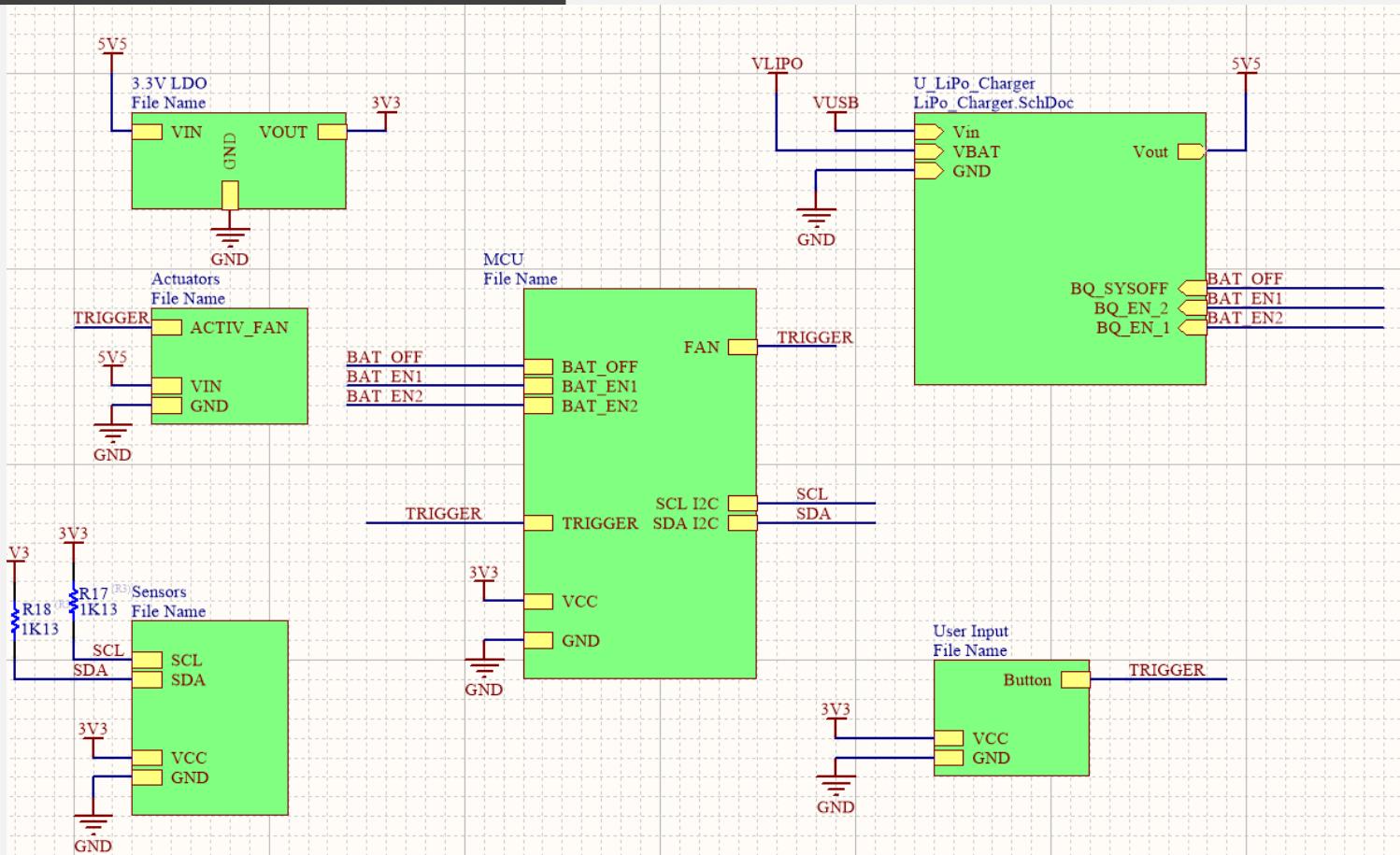


- A work flow from left to right (in general) is preferred.
- We use connectors to help drive the flow of the device.
- We can use nets and separate wires if it helps with readability.

CONNECTIONS AND NETS ON TOP SCHEMATIC

DON'T

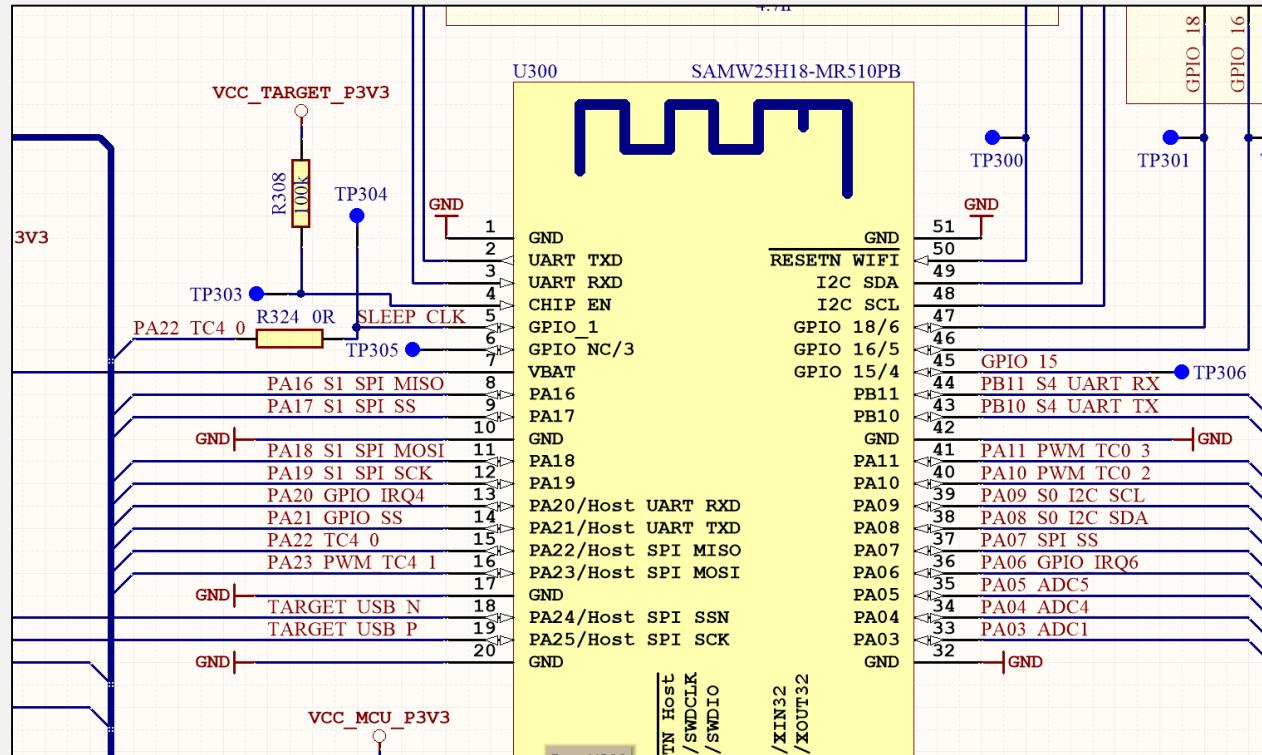
- Even if this is the same circuit as before, the flow of the program is lost because of the organization of the sheet symbols.
- Obfuscation in drawing can hide bugs! It can also confuse people who are helping you to review your board.



NET NAMES

- Pins on a Schematic Sheet are connected by lines / wires called **Nets**.
- Every Net in your design should be labeled with a **Net Name**.
- This makes identification during PCB routing much nicer, as well as more advanced schematic capture.
- It's good practice for industry!

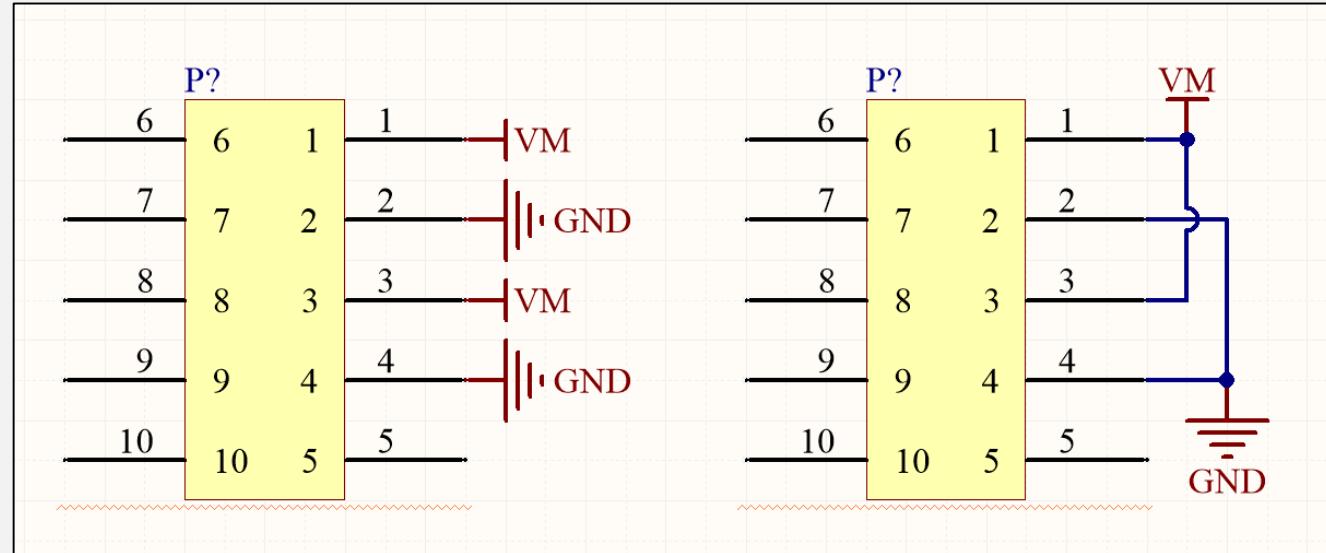
DO



POWER PORT ORIENTATION

- For clarity, always attempt to orient GND down and point positive voltages up.
- Sometimes, it can be clearer to orient GND or VCC sideways, which is acceptable.

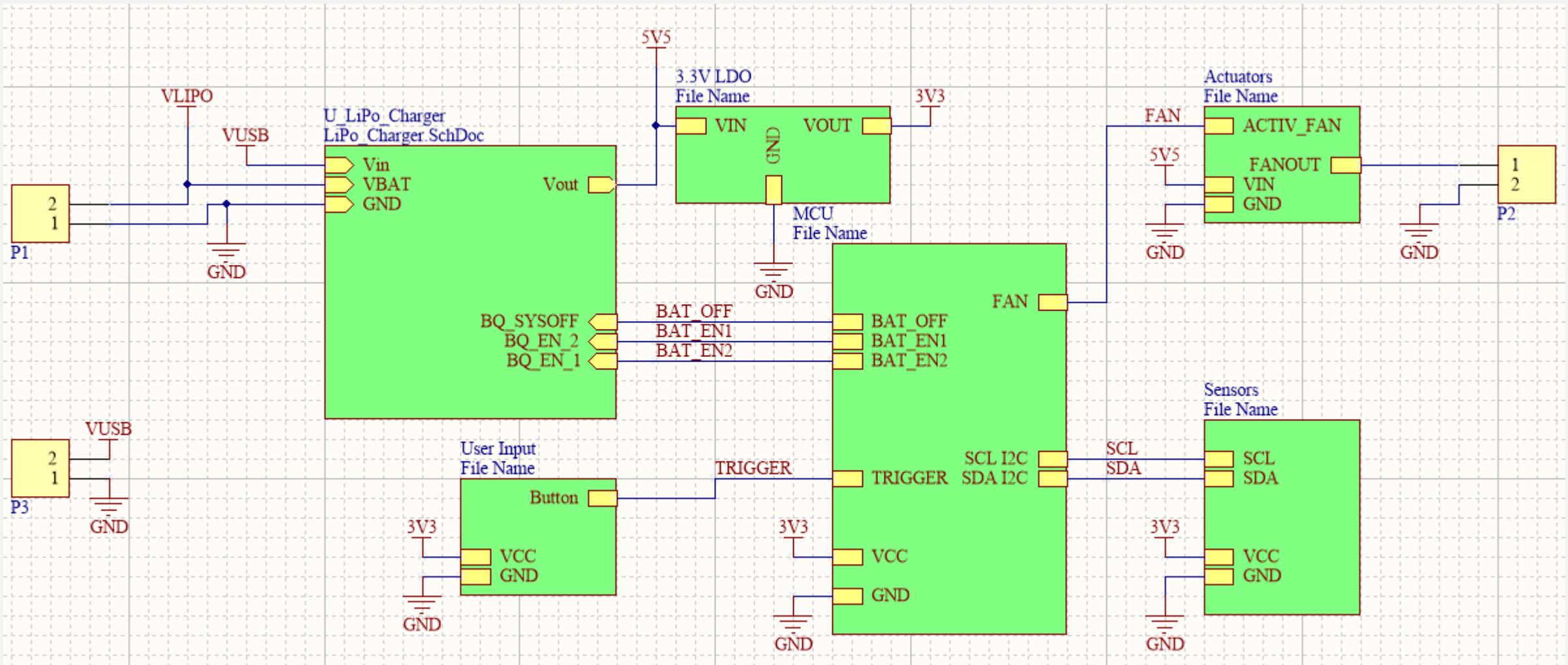
DO



DIVIDING YOUR DESIGN

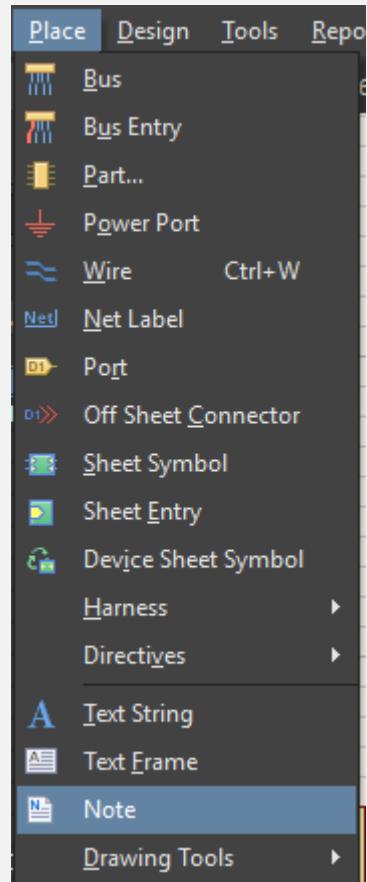
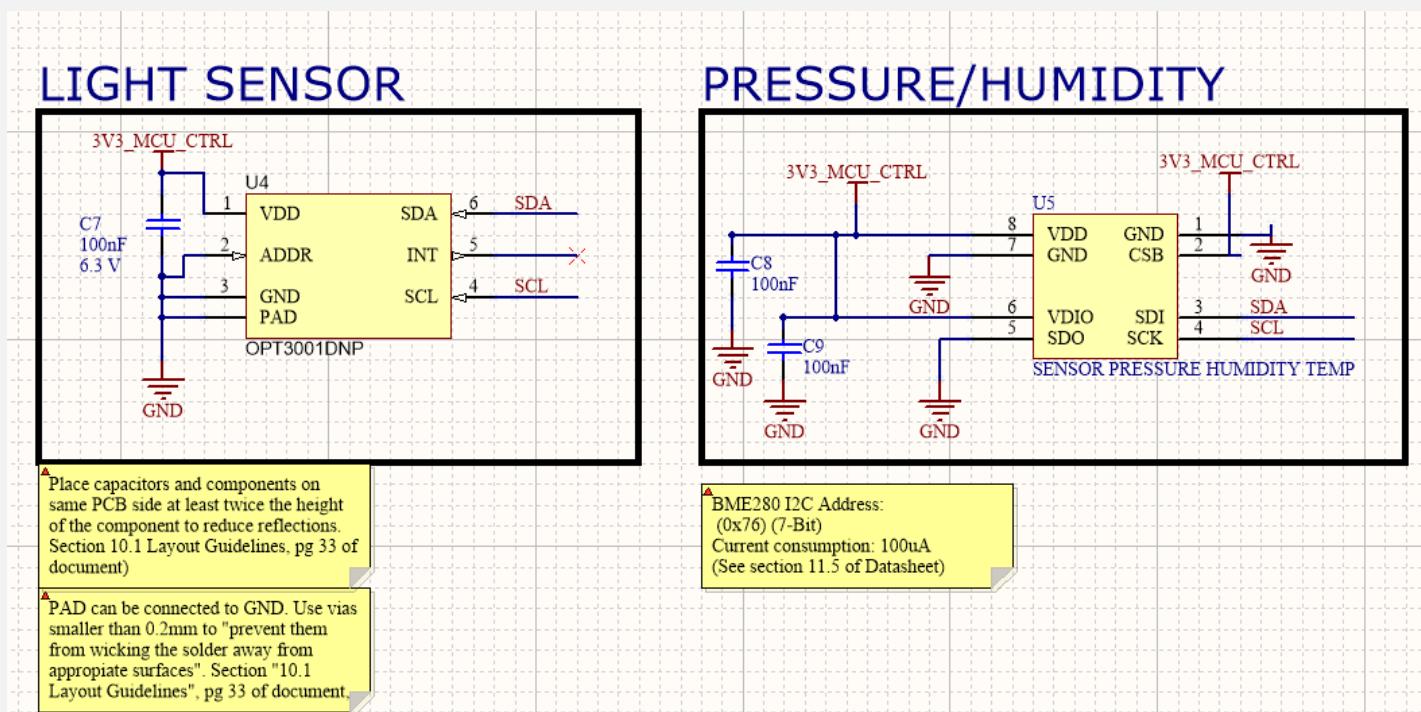
- Try to divide your design by grouping up components into sheets that have similar function.
- Have always a separate sheet for the Microcontroller (MCU).
- Groups sensors with respect to function. If you have a small quantity of sensors, you can put them on a “sensors” sheet.
- Same thing for actuators!
- Connectors into the system go on the TOP SHEET. No buried connectors please!
- Divide the power circuitry on their own sheet

DIVIDING YOUR DESIGN



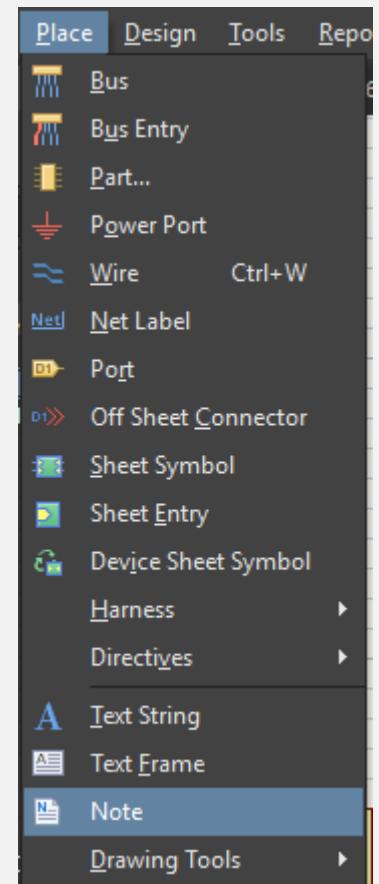
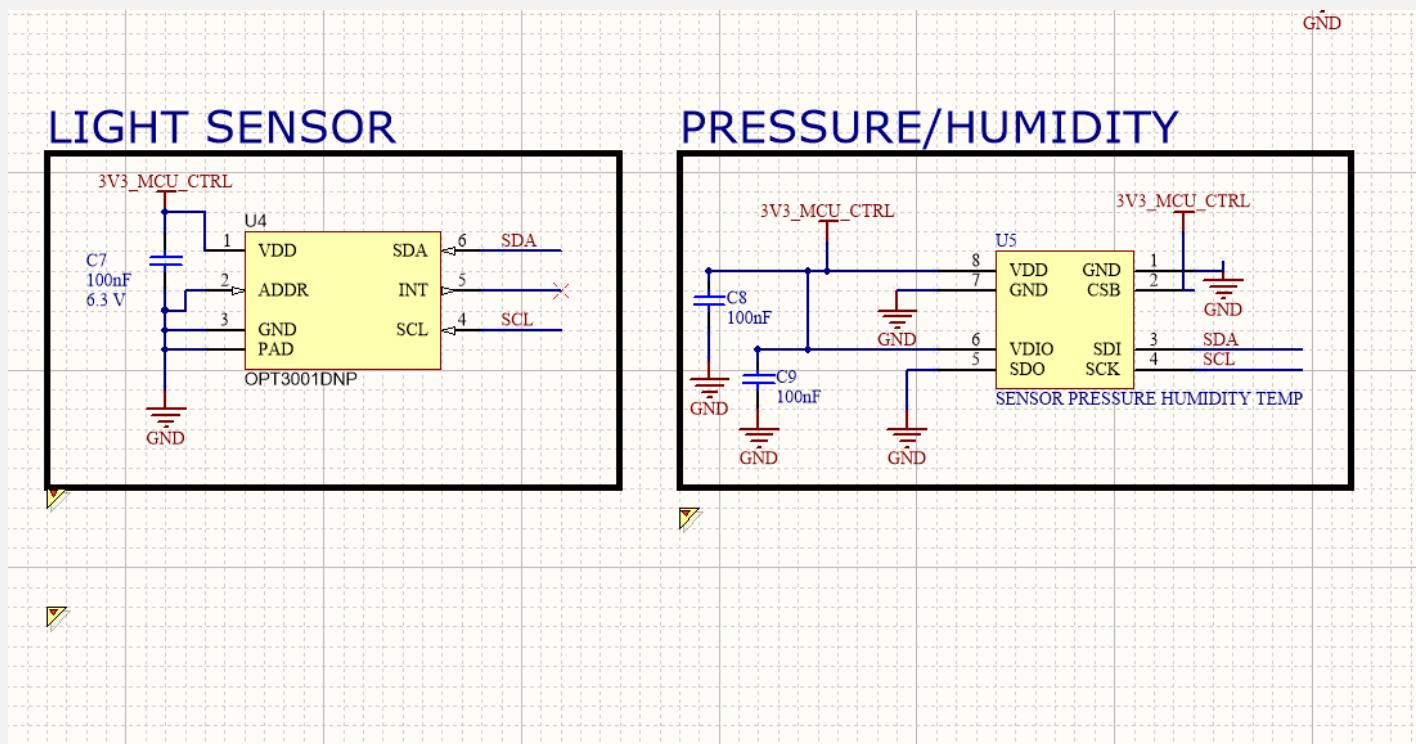
DIVIDING YOUR SHEET

- Organize similar circuitry inside your sheet
- Feel free to use graphical boxes and text to group up parts of your schematics and explain their function. But don't overdo it!
- For I2C Components: I recommend to write down their address so it is easy to use in code later!
- Please add comments (Place -> Note). This adds a note that can be minimized.



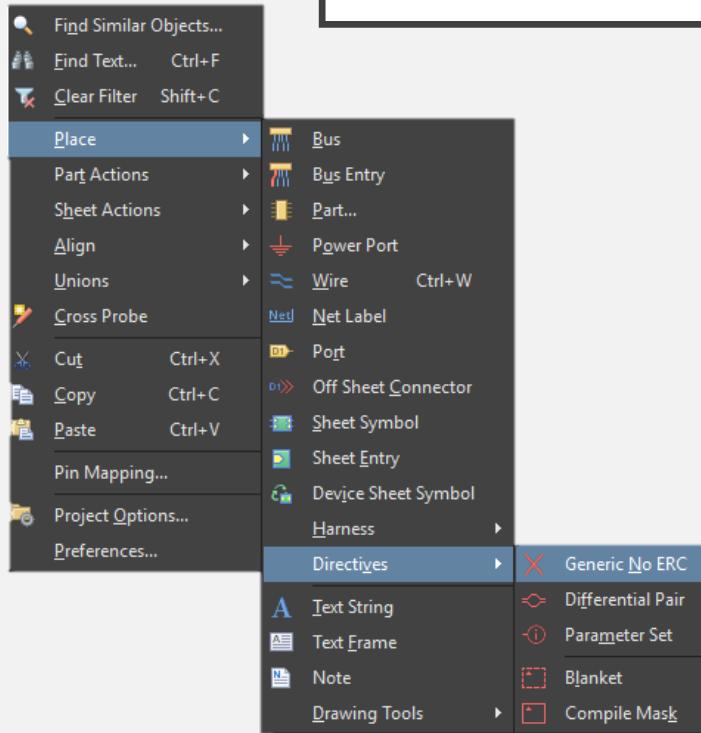
DIVIDING YOUR SHEET

- Organize similar circuitry inside your sheet
- Feel free to use graphical boxes and text to group up parts of your schematics and explain their function. But don't overdo it!
- For I2C Components: I recommend to write down their address so it is easy to use in code later!
- Please add comments (Place -> Note). This adds a note that can be minimized.

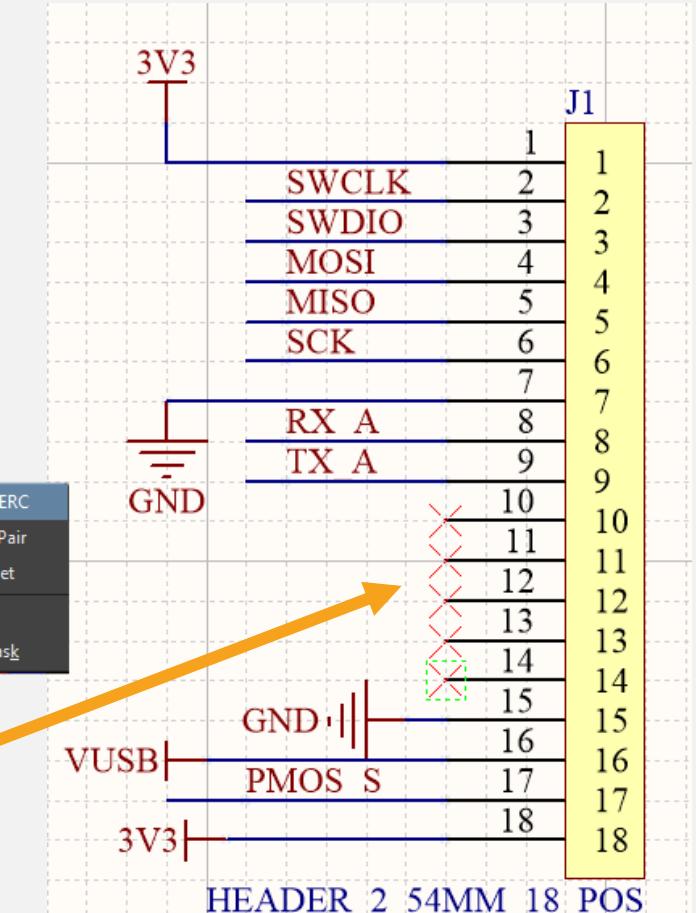


USE THE NO ERC DIRECTIVE

- Directives are used to denote important items on schematic sheets
- No ERC directives suppress warnings and errors where you place them
- Use the Generic No ERC (a red X) when you are not using a pin – this shows the reader that you intentionally did not route the pin.
- More info:
[https://www.altium.com/documentation/19.0/display/ADES/Sch_Obj-NoERC\(\(No+ERC\)\)_AD](https://www.altium.com/documentation/19.0/display/ADES/Sch_Obj-NoERC((No+ERC))_AD)



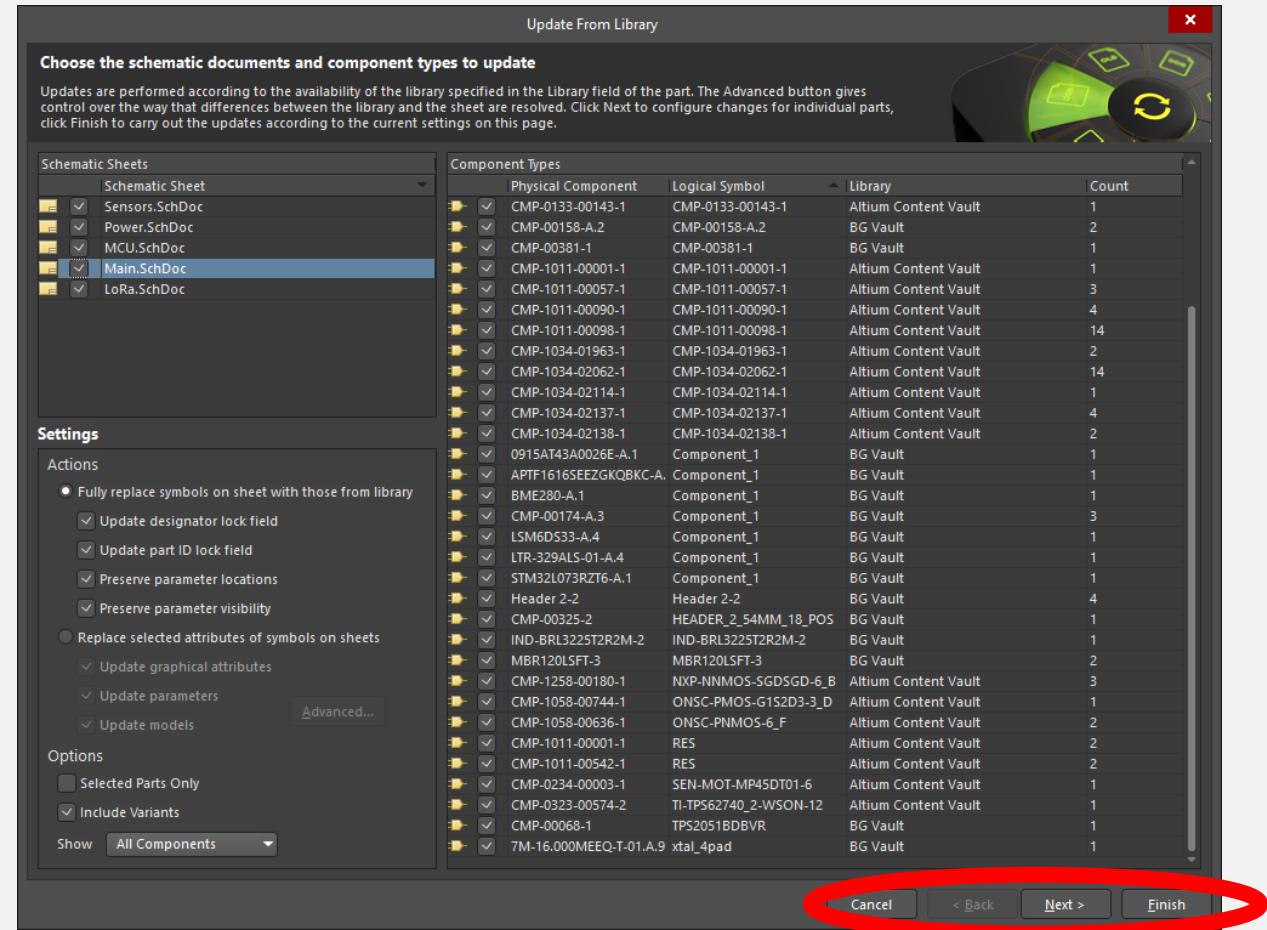
Tells Altium we are not connecting these pins to anything



DO

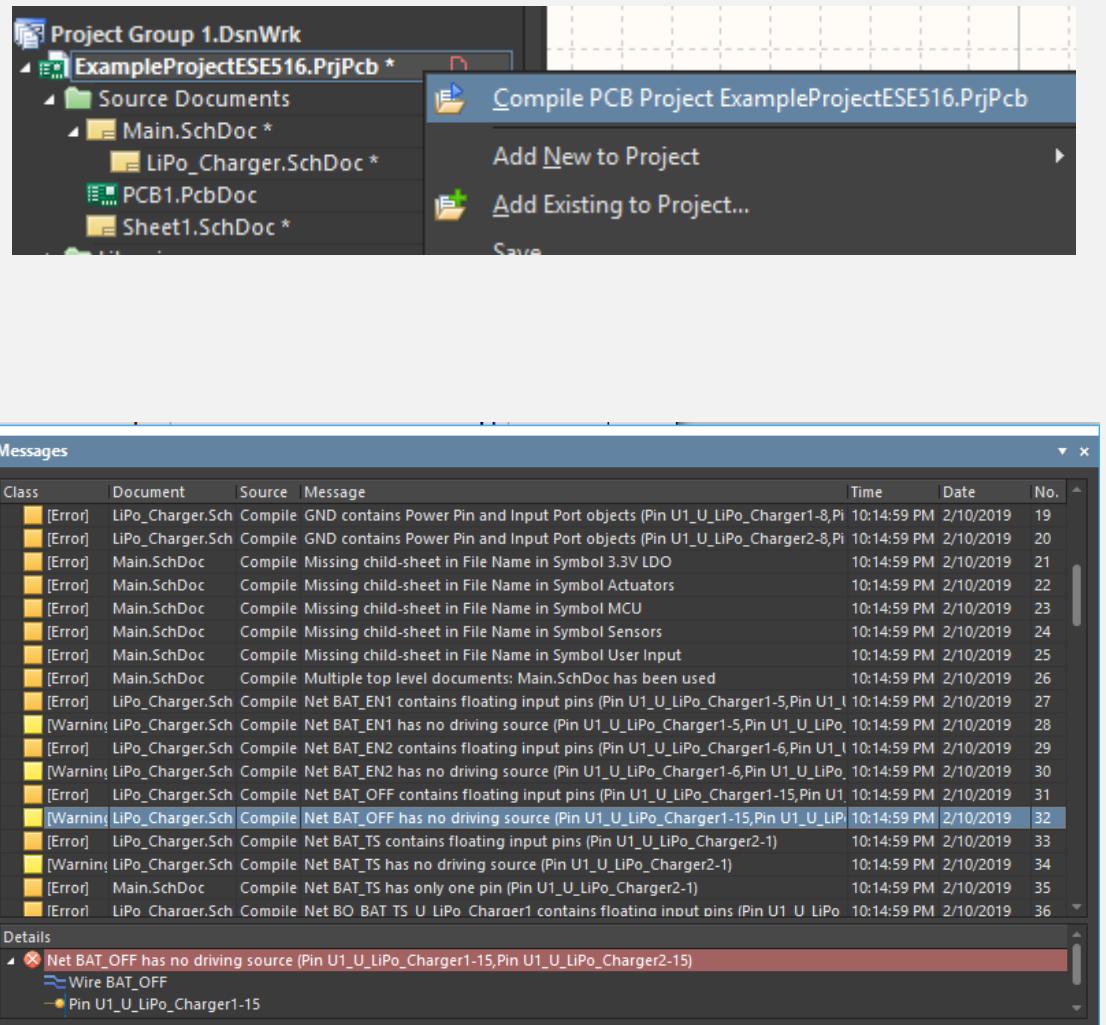
UPDATING COMPONENTS IN SCHEMATICS

- If you update components in your Schematic Library, they will not automatically change in your schematic sheets.
- You can either delete and replace manually, or you can do a sweeping change of all modified components by doing the following:
- **Tools > Update from Libraries...**
- Select all of the components you'd like to update. You can even only partially replace components – for example, if you only changed the part description.
- Click through, execute the ECO, and you're done.



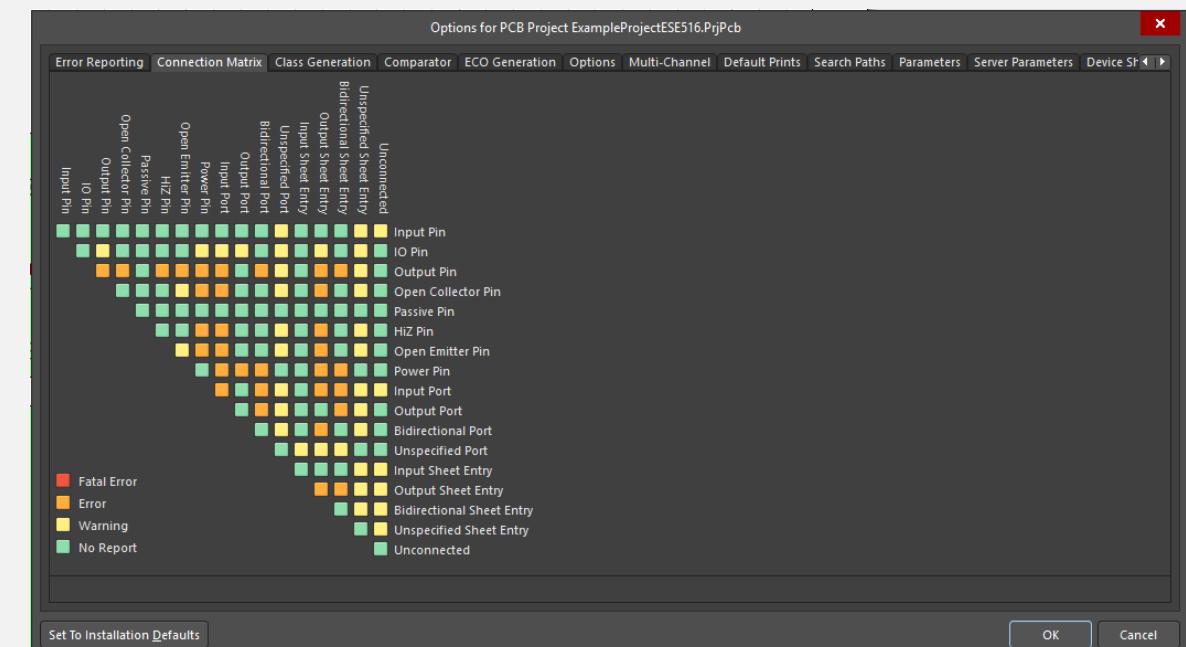
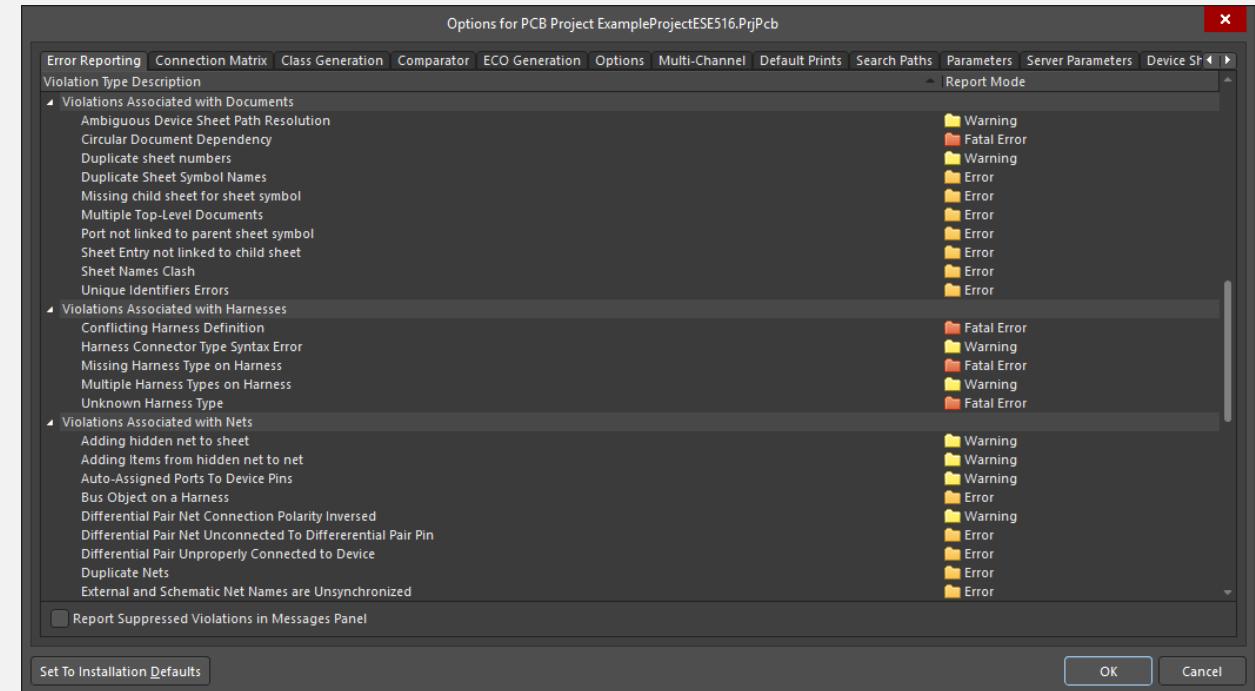
COMPILE PROJECT

- Altium projects can and should be compiled to determine errors in Schematic Capture.
- Common issues include:
 - Floating input pins (turn off)
 - Input connected to input, output to output (ensure this is correct wiring, and that you made the schematic symbol correctly)
 - No driving source (turn off)
 - Components are not named, duplicate names (name or rename them)
- Compiling messages are printed in the “Messages” panel.
- Try to figure out the issue yourself, and then ask a TA if you have trouble.
 - Also check out the next slide!



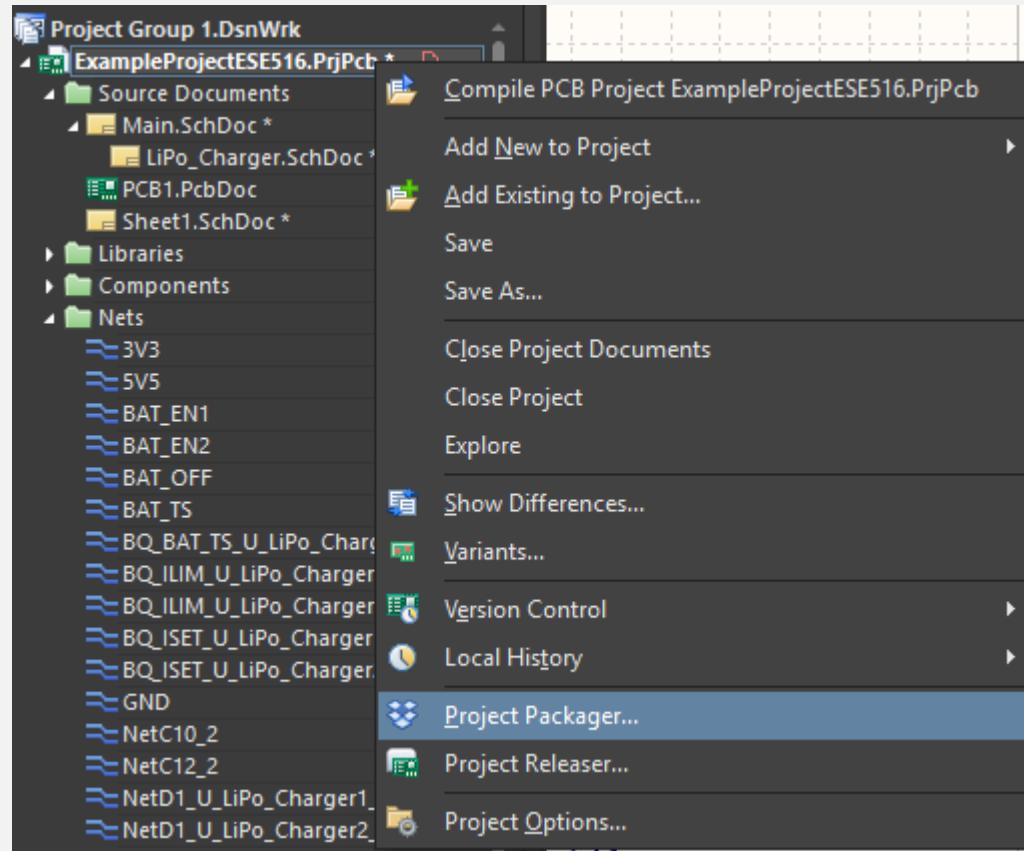
PROJECT OPTIONS

- Often, the default violations are not all suitable for your specific project. These errors and warnings can be modified by entering **Project Options**:
 - Project -> Project Options**
- Error Reporting & Connection Matrix tabs are relevant.
- Get familiar with the violations. It's not necessary to study in depth right now, but it's important to know what exists in the options.
- Generally, you'll come here after compiling and you determine that something considered an error is really a warning, or no report.



PROJECT PACKAGING

- The Project Packager is a great way of ensuring that all of your files are zipped together, retaining their links / relationships.
- The default options are fine – just click through and review, ensuring nothing funny.
- This is how you'll be required to submit all Altium assignments. Points will be taken away if the project is not packaged properly and unable to be viewed by the teaching staff.
- More information here:
[https://www.altium.com/documentation/19.0/display/ADES/WorkspaceManager_Dlg-WizardArchiver\(\(Project+Packager+Wizard\)\)_AD](https://www.altium.com/documentation/19.0/display/ADES/WorkspaceManager_Dlg-WizardArchiver((Project+Packager+Wizard))_AD)



CIRCUIT DESIGN BEST PRACTICES

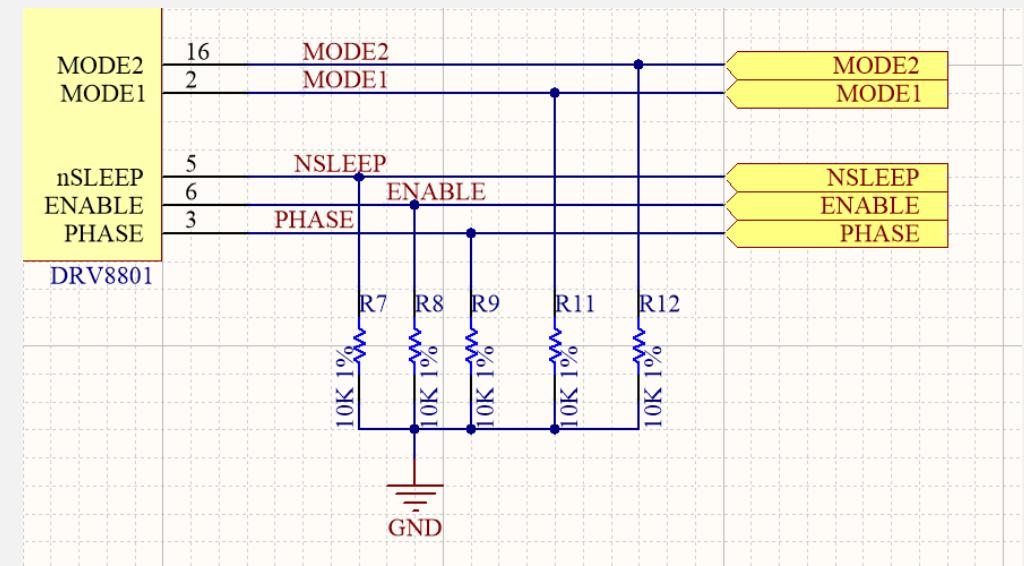
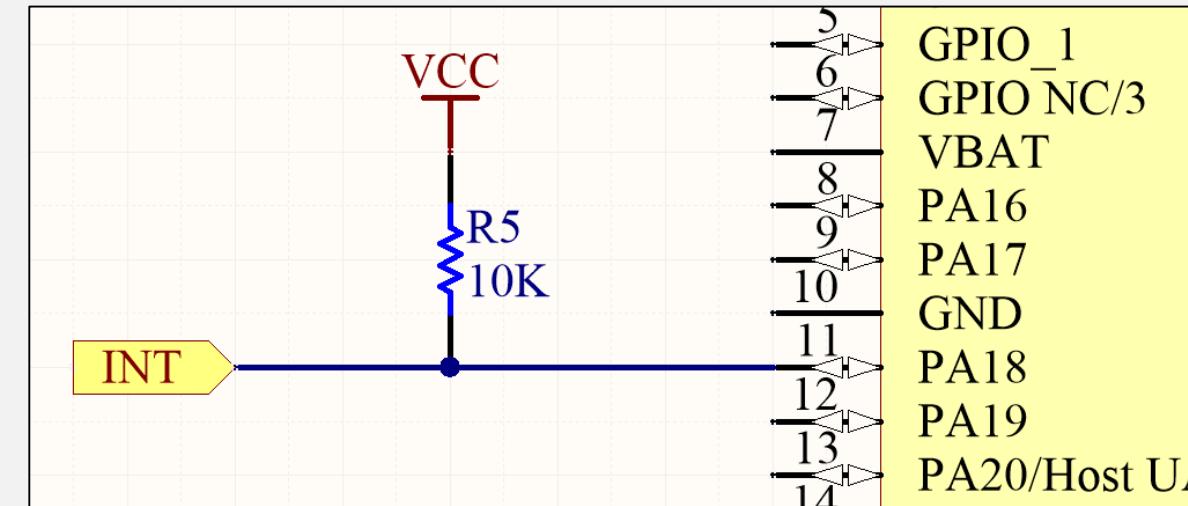
PULL-UP / DOWN RESISTORS

- **What do they do?**

- Pulls a signal wire to a known voltage / state
- Avoids the zone of uncertainty for digital states
- Important for power on & resets of the device, when pins may not be asserted on either the MCU or the peripherals

- **How do they do it?**

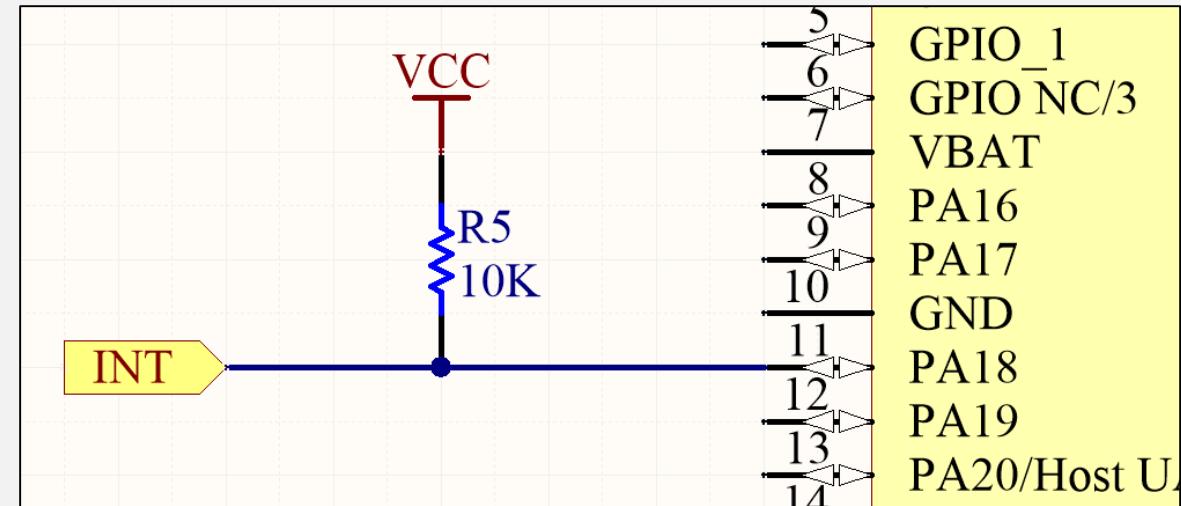
- “Weak” pull-up – a resistor that not a lot of current flows through, based on the wire length
- Should be strong enough to reject noise, but not strong enough that an IC cannot pull the line up or down
- Important to check IC details if uncertain – see what the pin drive current is, and calculate pull current
- Remember: Ohm’s law, so system voltage affects the strength of the pull up resistor current!
- $V = I \cdot R$, so $I = V / R \rightarrow I = 3.3V / 10k\Omega = 0.33mA$



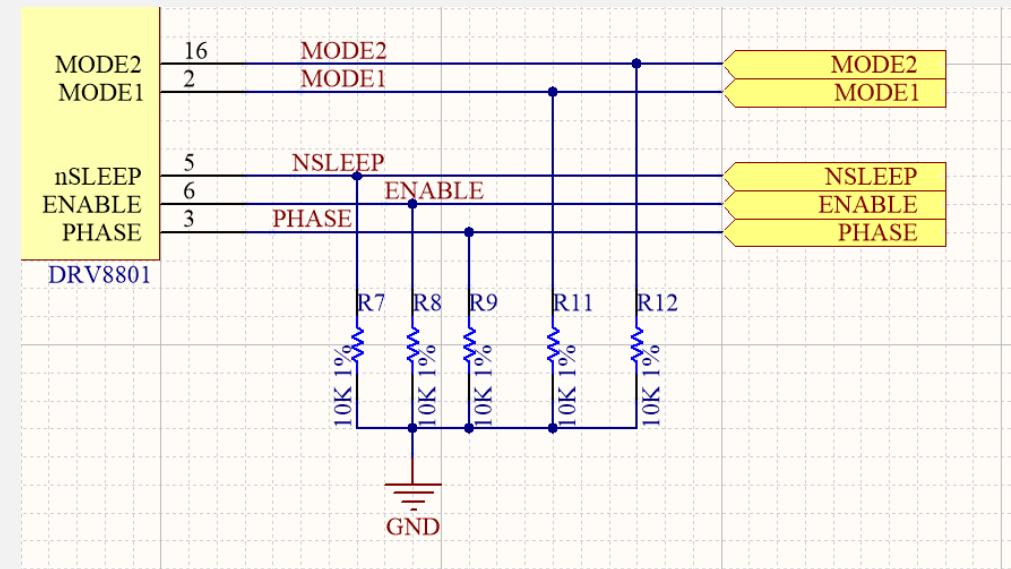
Example: Pull down motor driver input to GND (default setting)

PULL-UP / DOWN RESISTORS

- **Where should I use them?**
 - Any pins driven by interrupts
 - Buttons
 - Reset pins
 - You want to set the default input into an IC
(Example: Motor driver, don't activate as default!)
- **What happens if I don't use them?**
 - Unknown states
 - Ghostly MCU or devices resets
- **Where should I put them?**
 - Put them on the schematic of the IC that needs them!



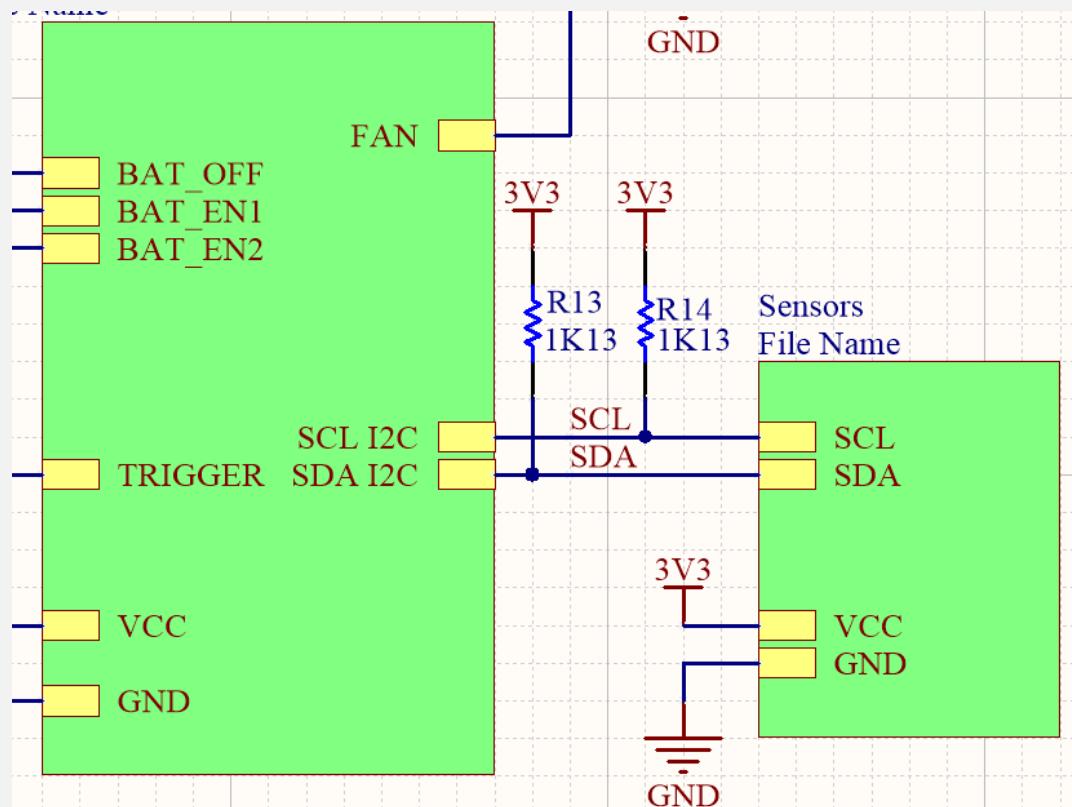
DRV8801



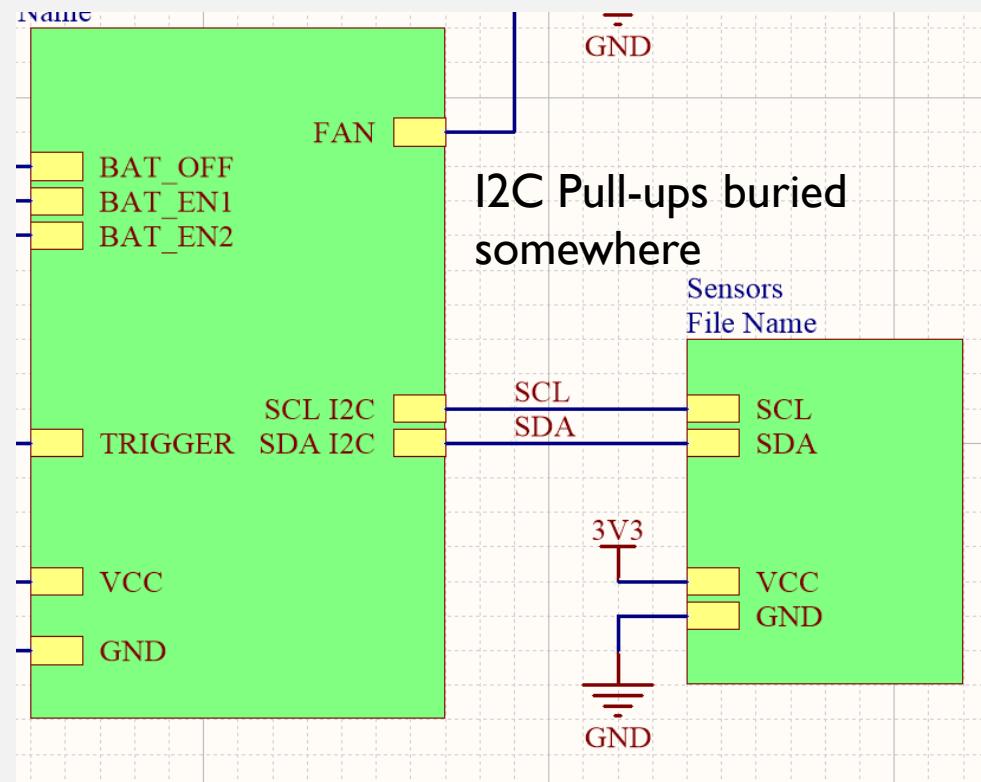
SPECIAL CASE PULL-UPS – I2C

- I2C pins are Open Drain. Each pin needs a pull-up to operate. It is good practice to put the I2C pins on the TOP SHEET. It is bad practice to “bury” the I2C pins inside a sheet!

DO

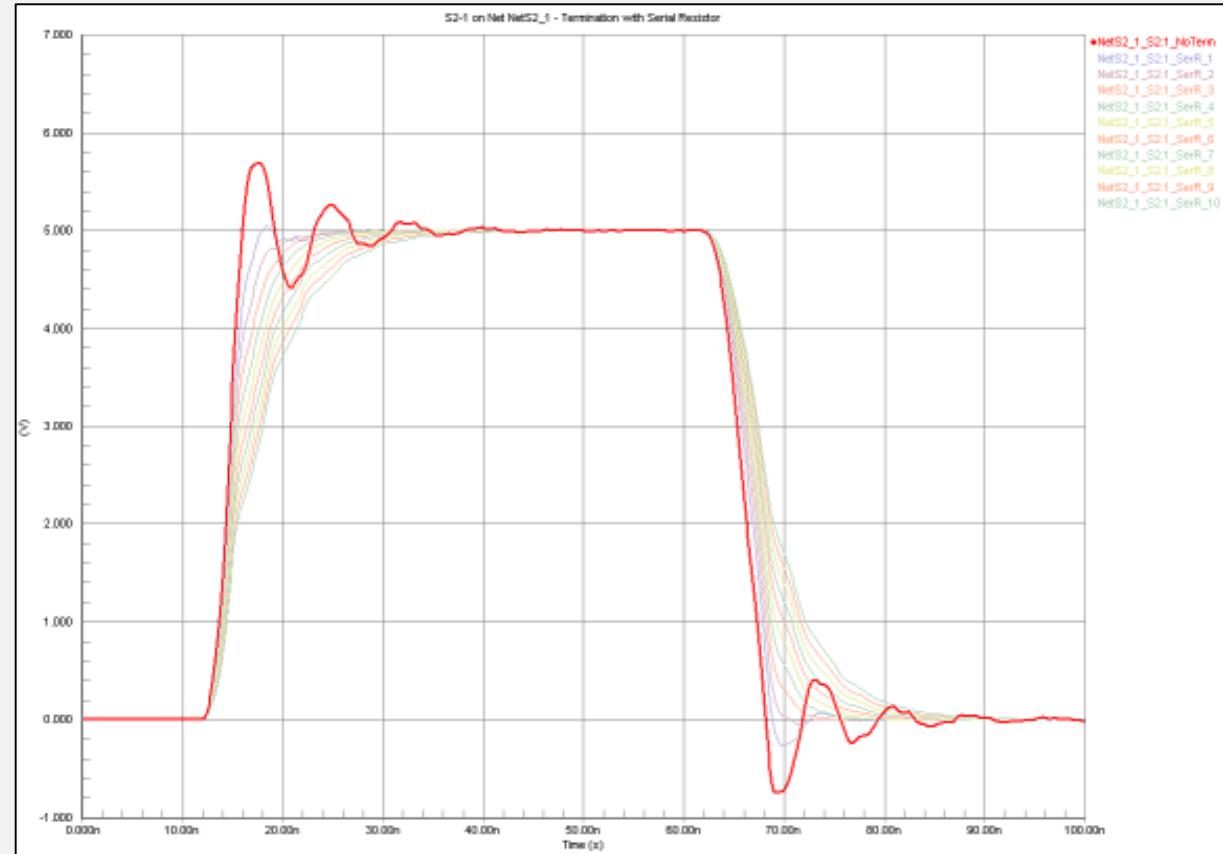


DON'T



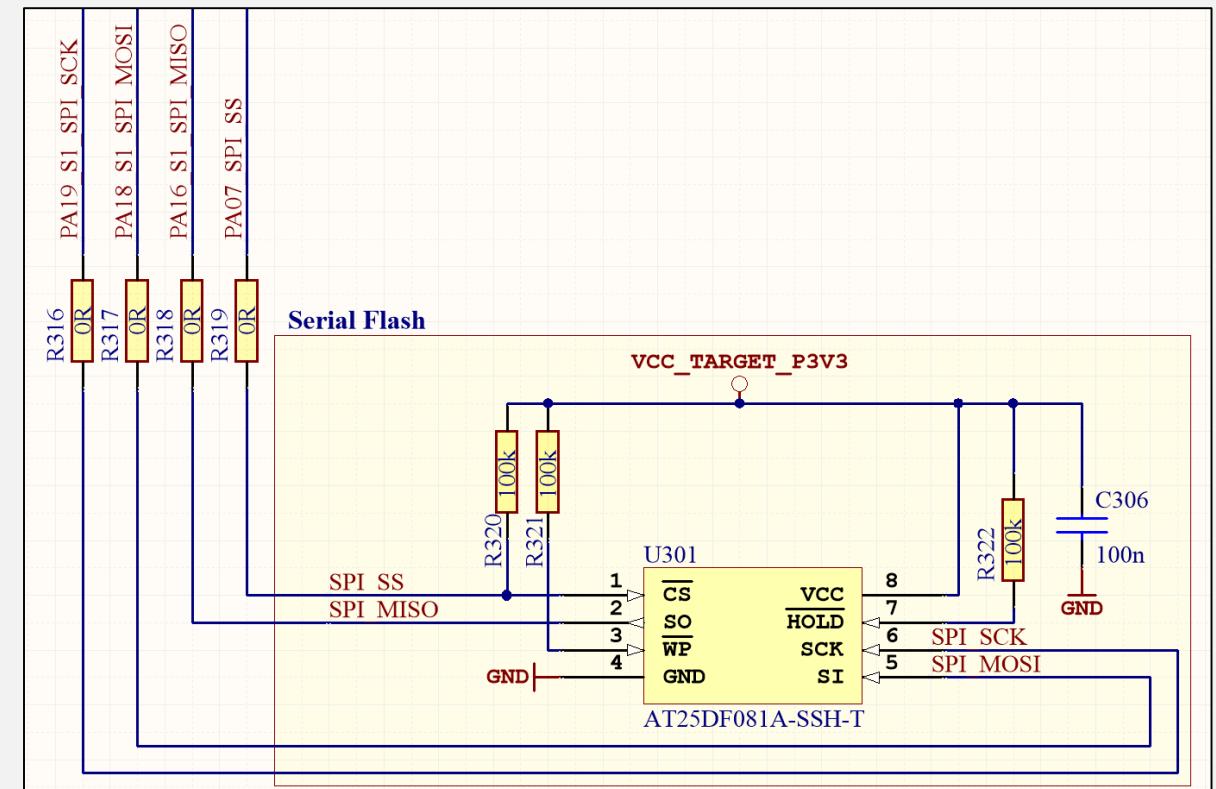
SPI -SERIES TERMINATING RESISTORS

- Every digital connection should be considered a transmission line with a non-instant connection
- Traces are a combination of inductance & capacitance
 - So, it's an LC circuit
 - Short traces = shorter circuit = less effect on the trace
 - High frequency lines need to consider this – be concerned if you're in the MHz, but the effect is more pronounced at higher frequencies
 - Longer traces exacerbate the issue
- The interface between LC circuit and “open” results in ringing
- Used in cases where high speed digital signals can be reflected and cause noise – like your SPI lines



SPI -SERIES TERMINATING RESISTORS

- **How do I fix it?**
 - Put resistors in line with the signals
 - Ex: For SPI, you'd be worried about CLK, MOSI, MISO – running at 12MHz max clock rate for SAM D21
 - Ex: I2C runs at a max of 400kHz, don't need to worry about it
- Resistors go close to the **driver side**, not the receiver side
 - So, MOSI = Master Out Slave In, so it should be close to the Master pin.
 - MISO would be the opposite, Master In Slave Out, be closer to Slave pin.,
- Resistor value depends on PCB, trace width
 - But, it'll be low – under 200 ohm
- Learn more
(<http://electronics.stackexchange.com/questions/10884/how-termination-resistors-work-what-happens-if-i-use-lower-values>)

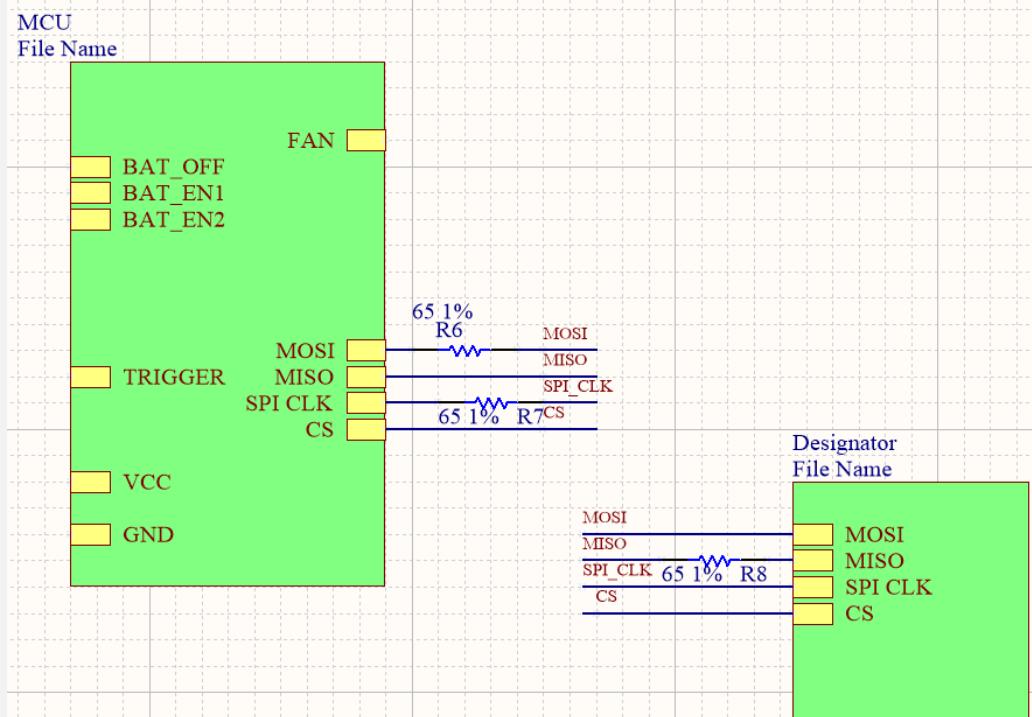


SPI-SERIES TERMINATING RESISTORS

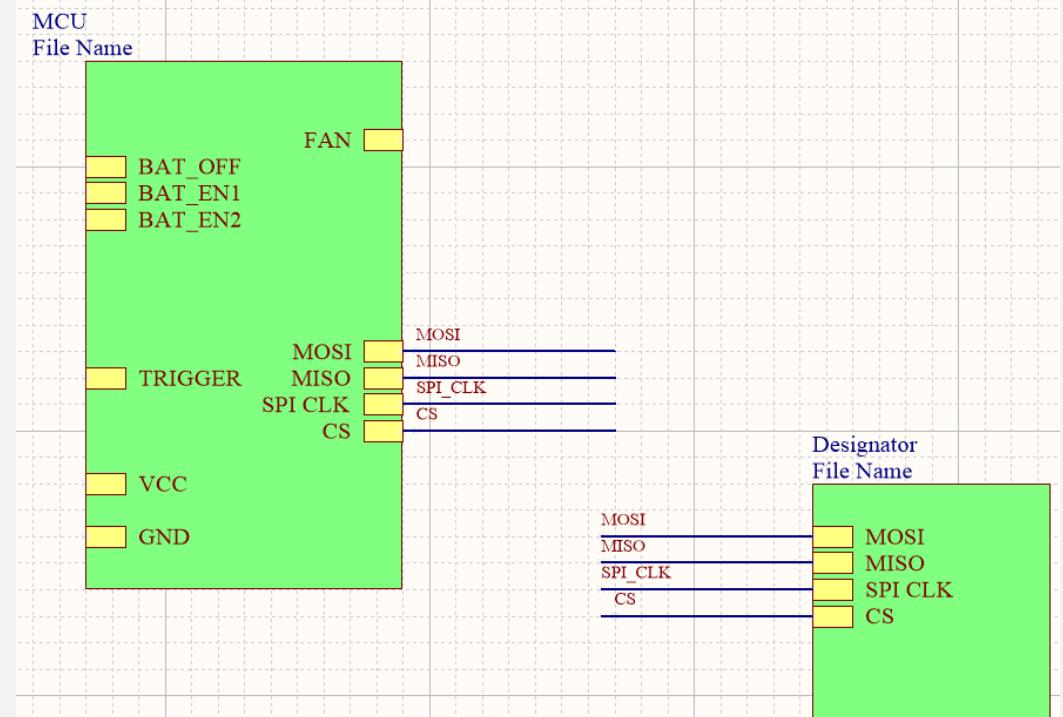
- **How do I draw it?**

- Similar to I2C, place the series terminating resistors near the source on the TOP sheet.

DO

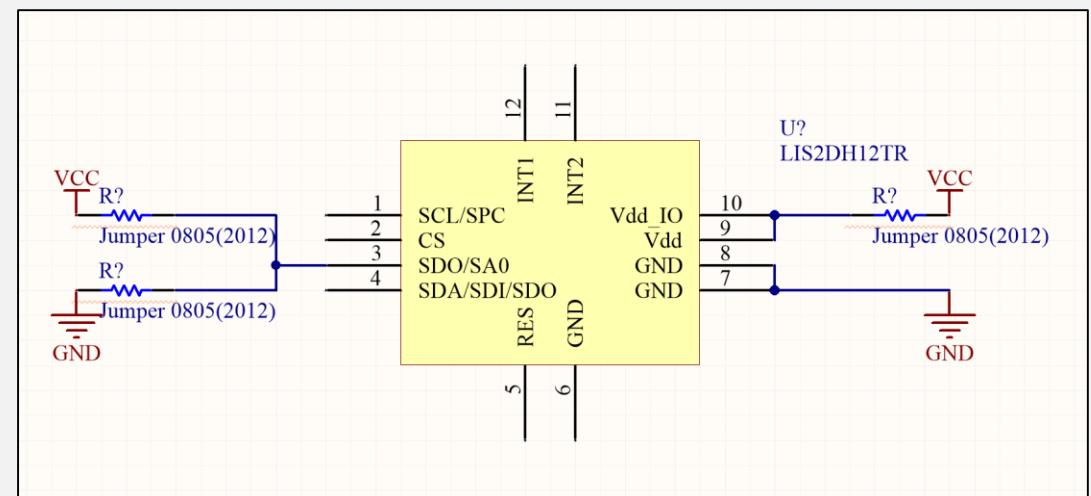
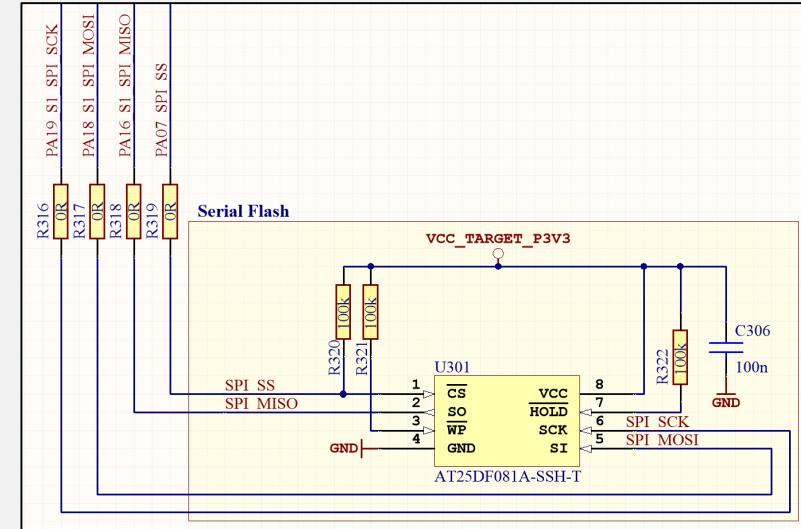


DON'T



DEBUGGING JUMPERS

- Handy for testing PCB section by section
 - For the first board bring-up, it's smart to validate your power works before hooking it up to peripherals, potentially damaging them.
 - Ex: Switching regulator runs high, outputs 5V instead of 3.3V
- In-line current measurement
 - Isolate sensors, actuators, WiFi, MCU to determine expected vs. real current draw
- Swap protocols for a sensor / actuator
 - Ex: Switch between I2C & SPI protocols for an accelerometer
- I2C address setting

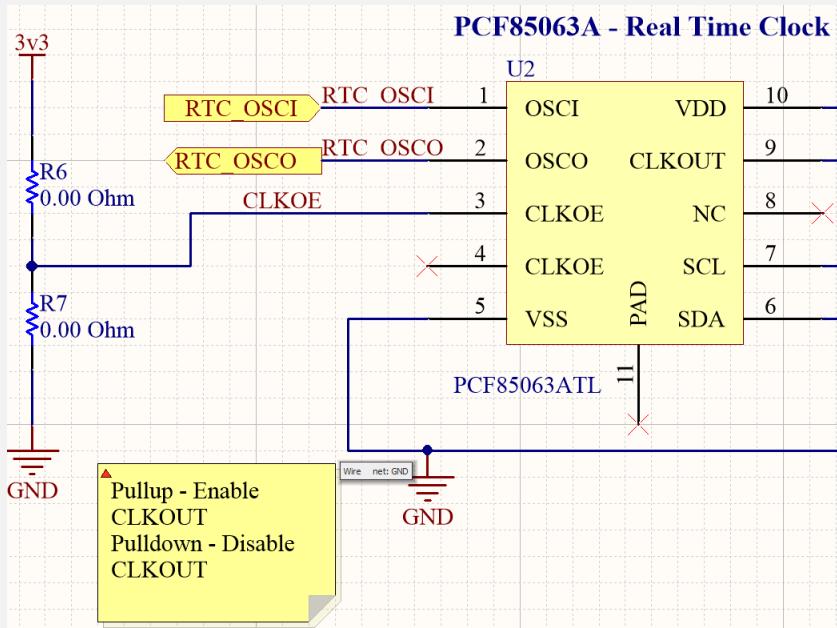


DEBUGGING JUMPERS

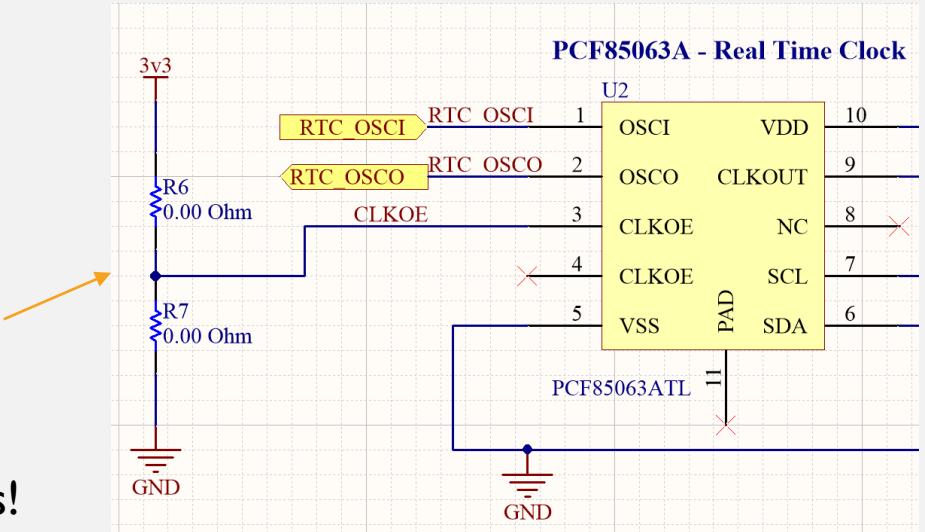
- **How do I draw it?**

- When adding debugging jumpers, it is an excellent idea to put a comment explaining what they do. We will learn later on how to decide which component gets populated.

DO



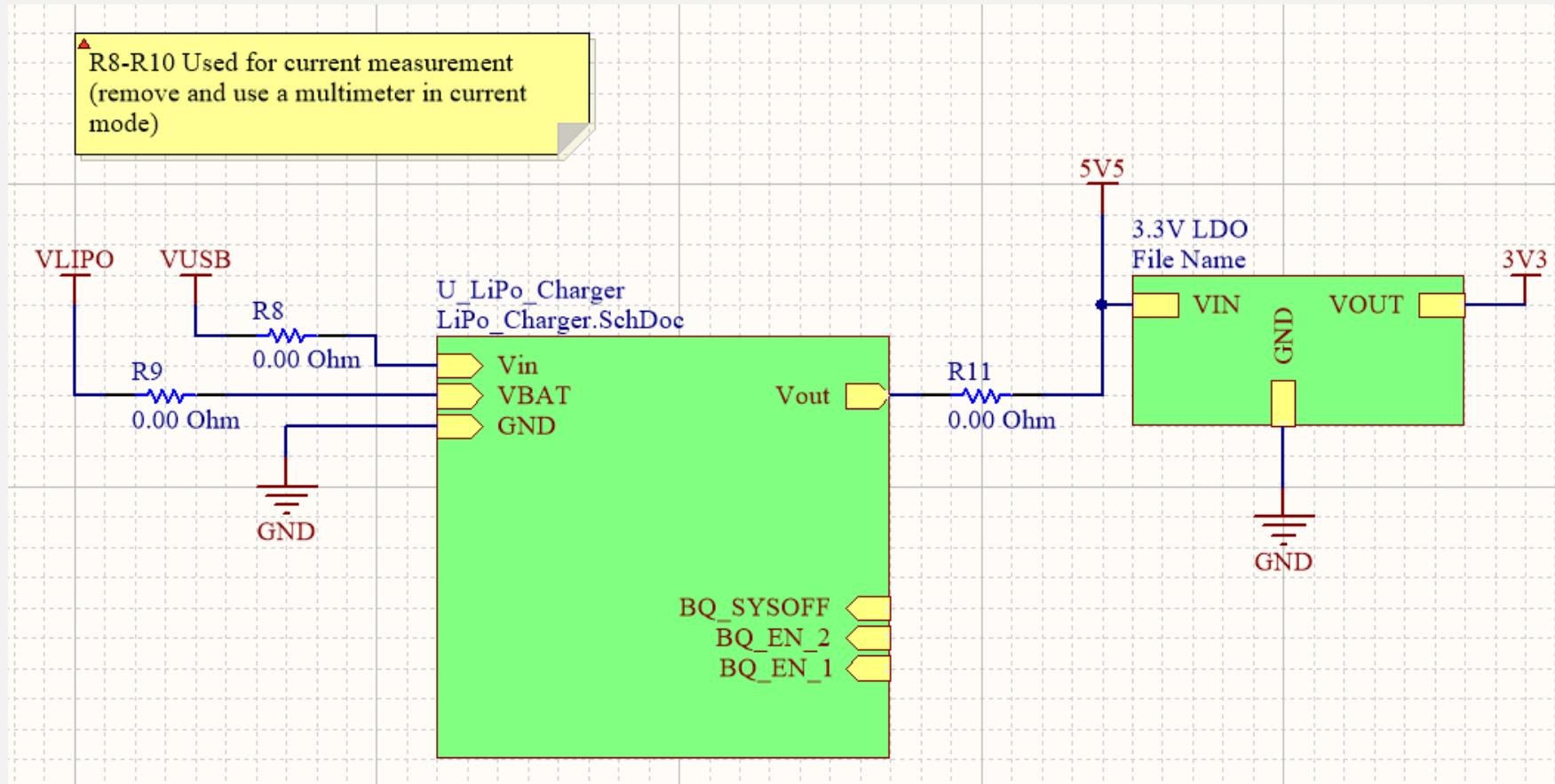
DON'T



What is
this for?
No
comments!

DEBUGGING JUMPERS – OTHER EXAMPLES

- Can be used to have a component where to measure current (Or disconnect subsystems for testing).

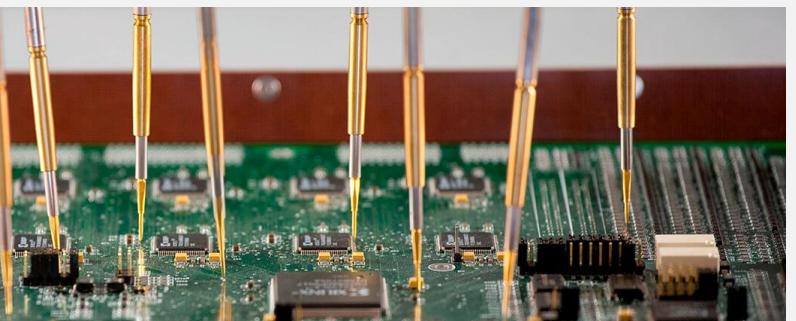


TESTPOINTS

- Testpoints are exposed copper pads (or a special component) that allow you to have easy access to a net in a PCB.
- As their name suggest, they are pads used for testing. They are very useful for debugging and checking the voltage of a pin
- They are also useful to debug communications!



Test point Components



Testpads for a Bed of Nails or Flying Probe

TESTPOINTS

- Add them to the most important nets:
 - Communications (I2C, UART, SPI, Etc.)
 - Voltage Rails (GND, 5V, LiPo, USB, 3.3V, etc.)
 - To MCU GPIOs that are important

Recommendations:

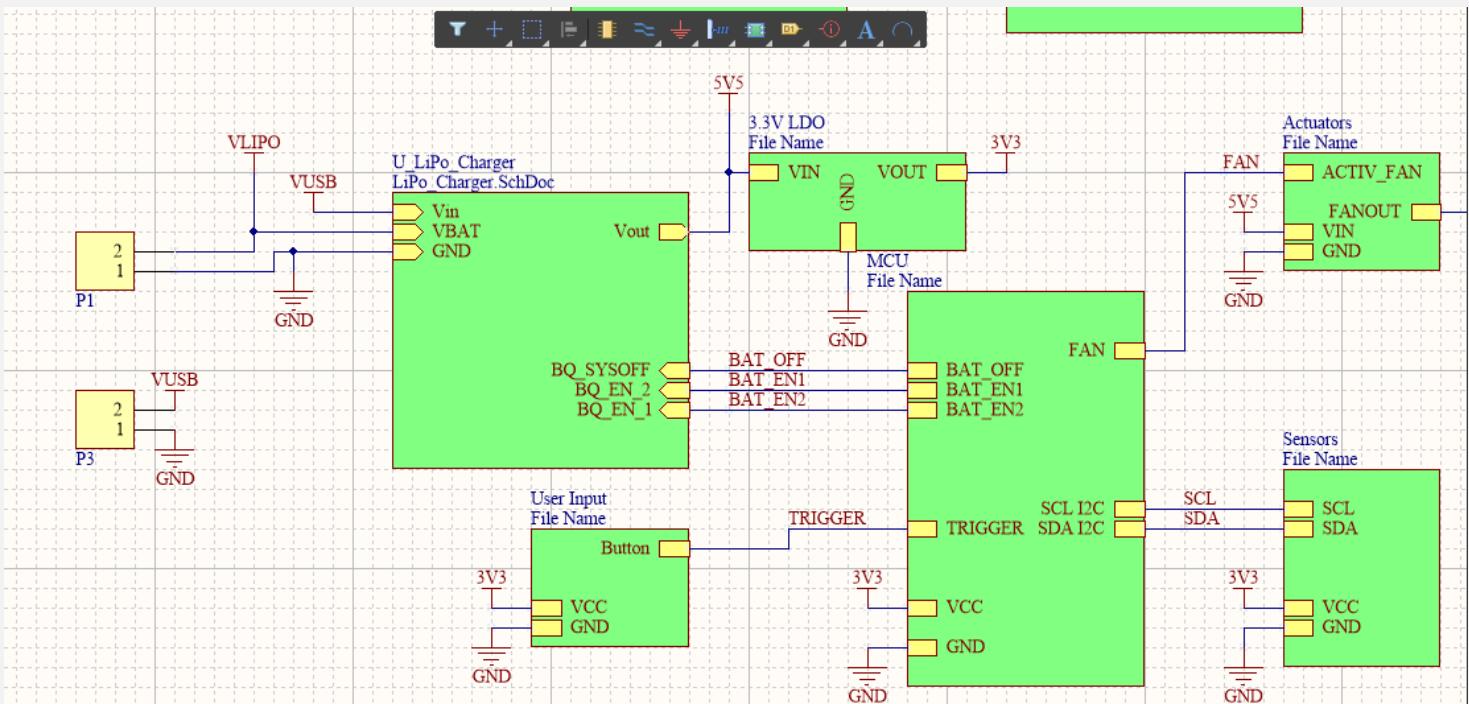
Space permitting, add Through hole test points for GND and power supplies (if you can, do at least for ground – it will make attaching an oscilloscope to your system way easier!)

Search for component 5011 on Altium Vault (or navigate to Unified Components/Components/Keystone/No Category for different testpoints)

For different nets, you can use the much smaller testpad (Search for 5019, in the same directory as before)

TESTPOINTS

- I usually add testpoints on a corner of the sheet where the net I want to test is present.
- In this example I added just a subset of the testpoints that would be adequate. I would recommend to have as a minimum:
 - All power nets, including ground
 - I₂C, SPI, UART, etc.
 - SWDIO, SWCLK (Programming pins of MCU)
 - Any GPIO that is important (in this example, the activation of the fan, or the trigger)



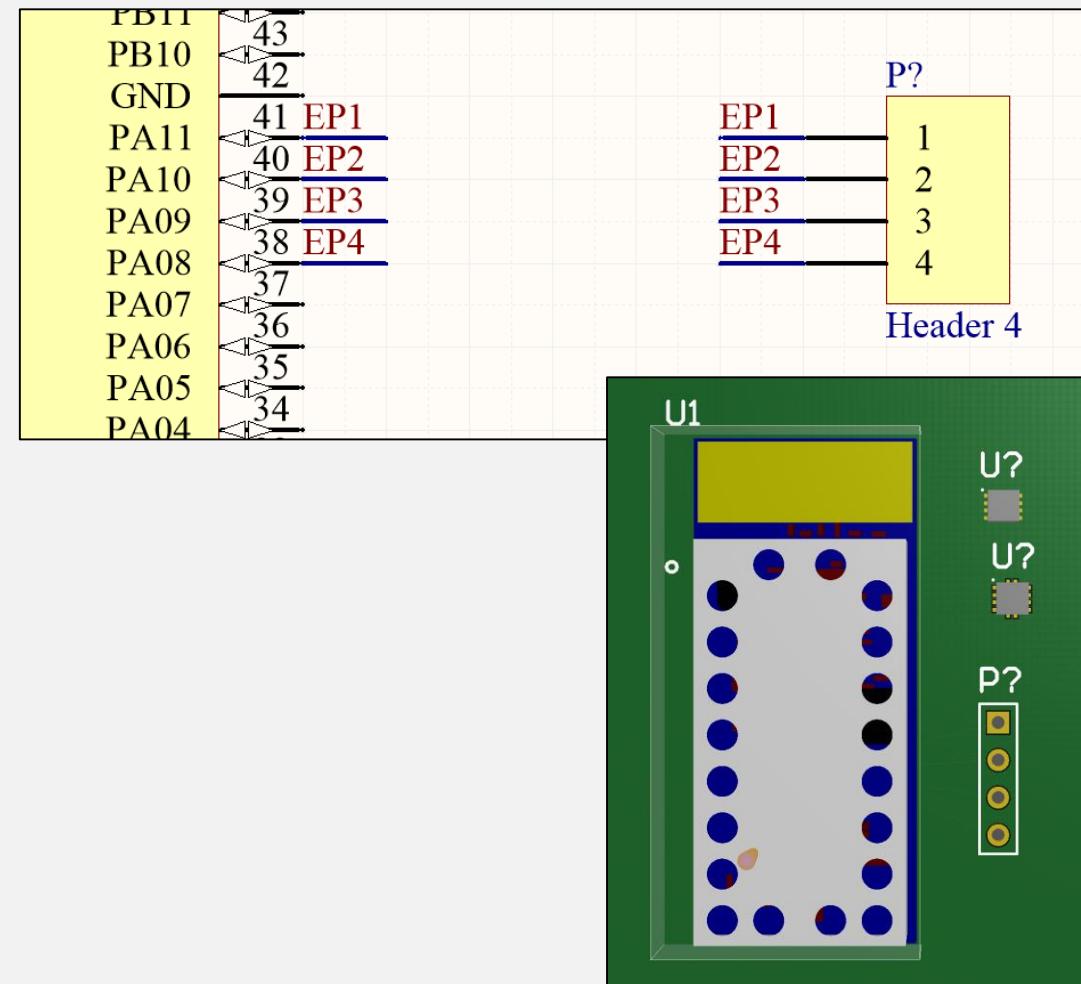
TESTPOINTS

TP1	SCL	
TP2	5019	SDA
TP3	5019	TRIGGER
TP4	5019	FAN
		3V3
		TP5
		5010
		TP6
		5011
		GND

Title	
Size	Number
A	
Date:	2/10/2019
File:	C:\Users\Main.SchDoc

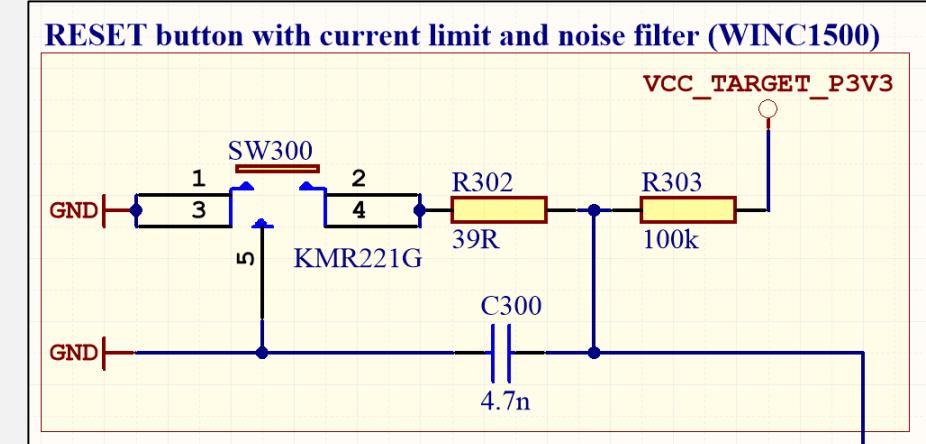
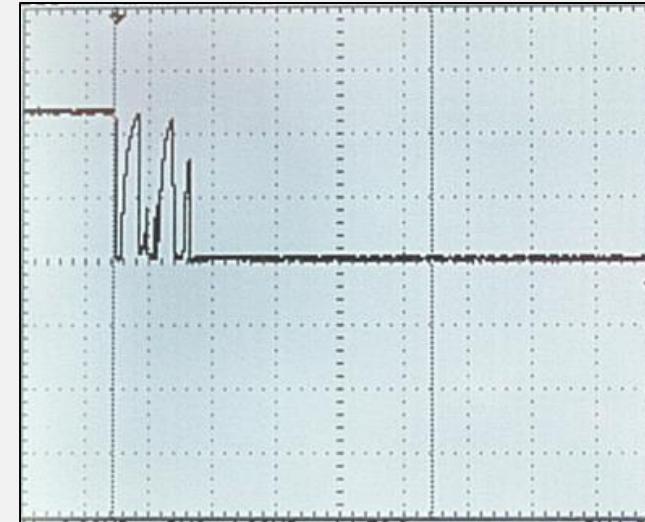
EXTRA MCU PINS

- Good practice in prototypes to pull out extra pins to pads or vias
- Gives an option for using these extra pins in the future – perhaps you want to add a sensor, or use it to read a debug signal
- Recommend to use a standard 0.1" header to pull out extra pins
 - If you don't have space for that, you can make a custom debugging pad that is not through-hole.



HARDWARE BUTTON DEBOUNCING

- Button signal lines can be really noisy – it's a mechanical issue
- Use resistors & capacitors to time a button debounce filter
 - Atmel SAM W25 source schematic has debouncing on the reset line
- This should be used in conjunction with firmware debouncing for best results
- [Learn more \(<http://www.golgi.io/debouncing-your-iot-buttons/>\)](http://www.golgi.io/debouncing-your-iot-buttons/)



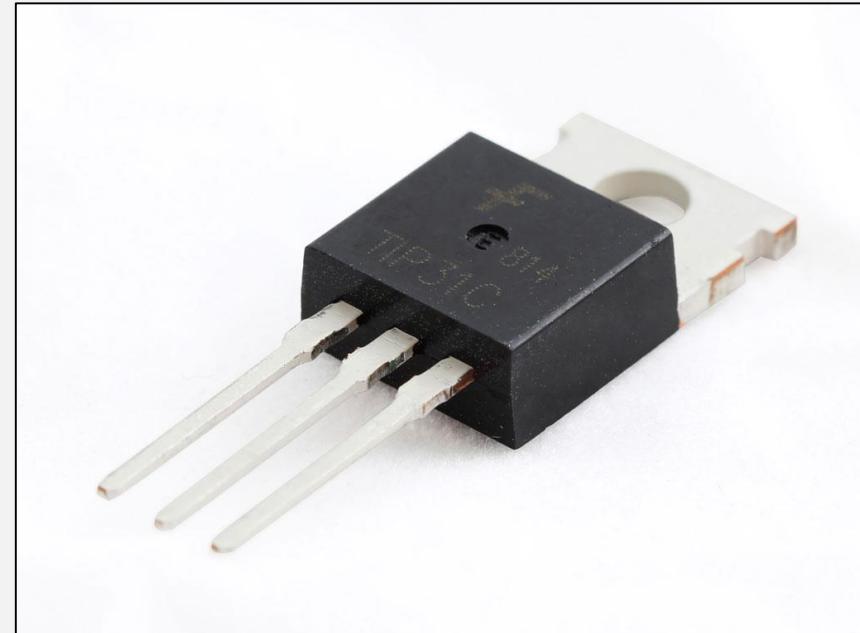
DRIVING HIGHER VOLTAGE, CURRENT OUTPUTS

- MCU pins can't drive everything
 - Typically, they can only sink or source something like 20mA (check datasheet)
- What happens when you need more current? Or want to drive a device at a higher voltage?
 - Transistors!

Table 4-6. DC Electrical Characteristics

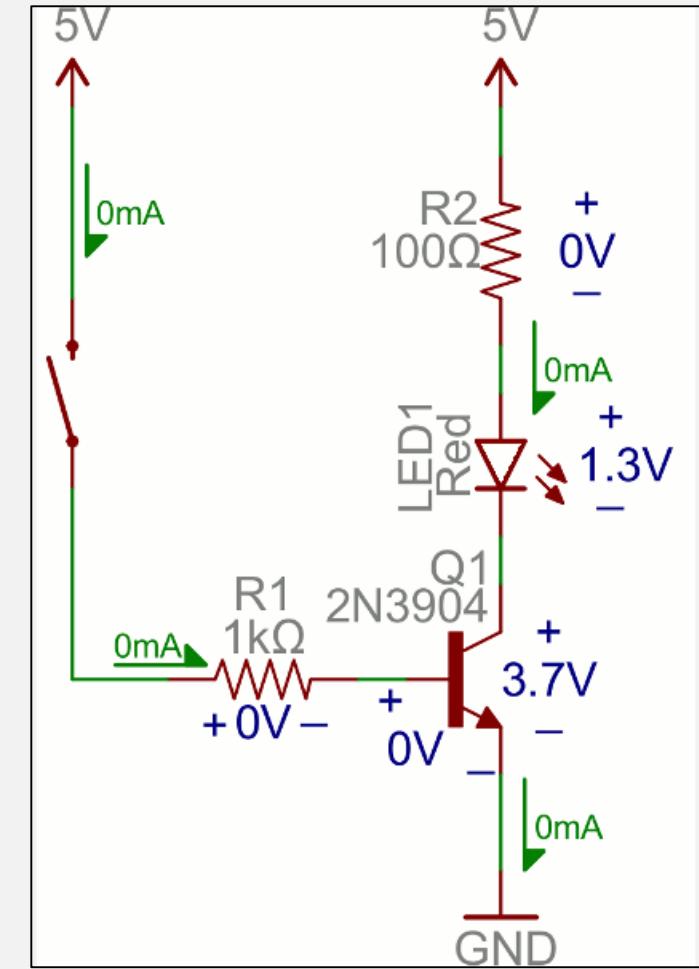
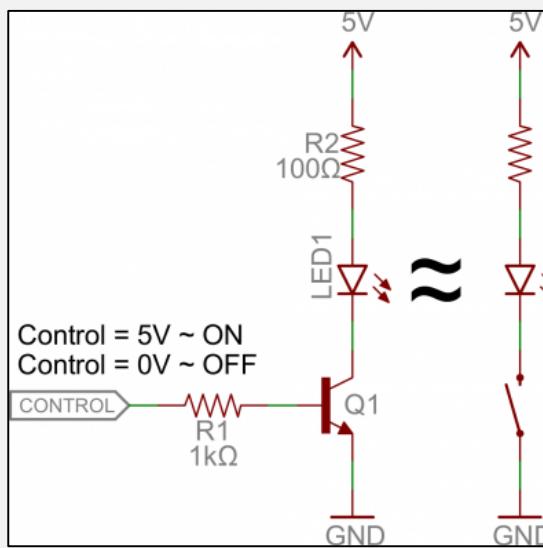
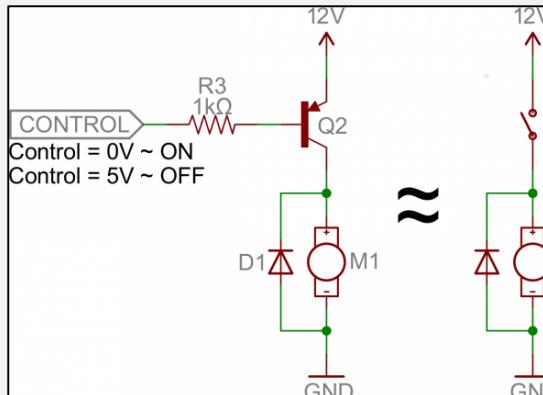
Characteristic	Minimum	Maximum	Unit
Input Low Voltage V_{IL}	-0.30	0.65	V
Input High Voltage V_{IH}	VCC-0.60	VCC+0.30	
Output Low Voltage V_{OL}		0.45	
Output High Voltage V_{OH}	VCC-0.50		
Output Loading		20	
Digital Input Load		6	pF
Pad Drive Strength (regular pads ¹)	8	13.5	mA
Pad Drive Strength (high-drive pads ¹)	16	27	

Note: 1. The following are high-drive pads: I2C_SCL, I2C_SDA; all other pads are regular.



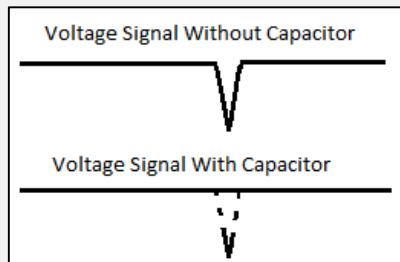
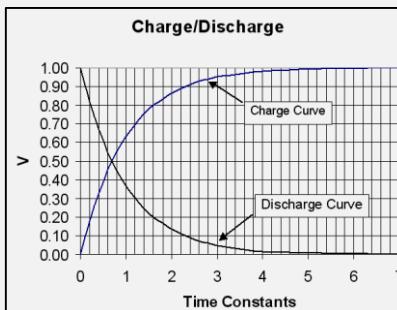
DRIVING HIGHER VOLTAGE, CURRENT OUTPUTS

- V_{th} (threshold voltage) defines what the voltage level needs to be for actuation
 - AKA – can a 3.3V logic signal turn on the transistor?
- High Side – PNP
 - To enable, logic must be higher than base voltage – could be an issue if the base voltage is high!
- Low Side – NPN
 - Can't flip the state of the device
- Current limiting resistor at the base
 - Can provide too much current and damage the component otherwise
- Learn more:
<https://learn.sparkfun.com/tutorials/transistors#applications-i-switches>



DECOUPLING OR “BYPASS” CAPACITORS

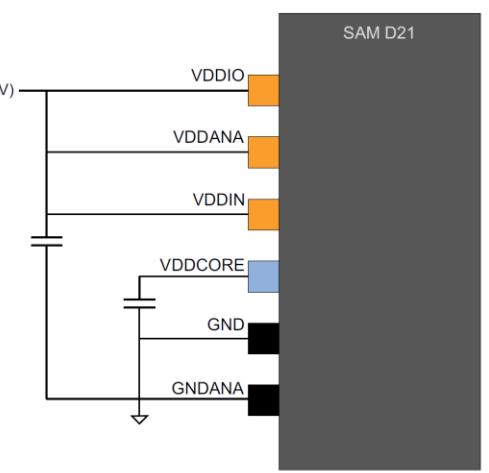
- **What is it?**
 - Local Energy Reservoir
 - Voltage drops – holds the line up
 - Spike on the voltage line – extra energy is absorbed
 - Shunted to ground
 - Passes only DC component & smooths signal out
- **What happens without them?**
 - In digital circuits – unexpected behavior. IC resets / brownouts
 - In analog circuits, noisy signals.
 - If audio, you can hear hums and crackling



6.2.3 Typical Powering Schematics

The SAM D21 uses a single supply from 1.62V to 3.63V. The following figure shows the recommended power supply connection.

Figure 6-1. Power Supply Connection



DECOUPLING OR “BYPASS” CAPACITORS

- **What size & type?**
- Important to handle high frequency spikes, as well as lower frequency ones
- Electrolytic capacitors are good for having a large charge storage, but have a slow response
- Ceramic capacitors have a quick response
- Typical to have 100nF / 0.1uF **ceramic** capacitors closest to the ICs for high frequency response
- Common to have a larger electrolytic / ceramic capacitor in the uF range for the PCB or circuit segment for low frequency response
- In general, **check the app notes!**
- App notes usually have typical schematic layouts for the IC, as well as PCB layouts
- The PCB layout can be critical for having low impedance paths for the current to move.

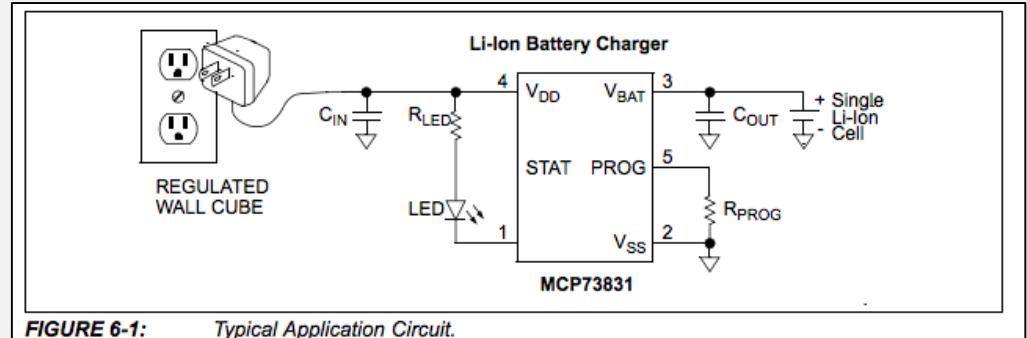
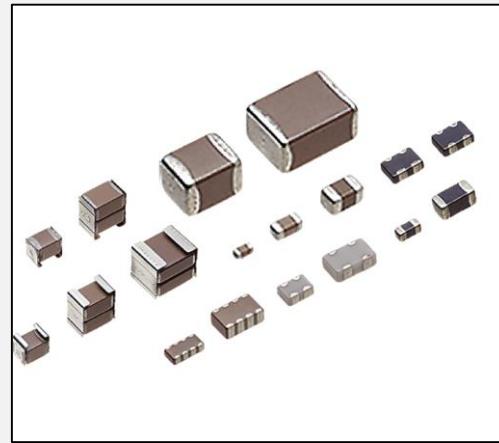


FIGURE 6-1: Typical Application Circuit.

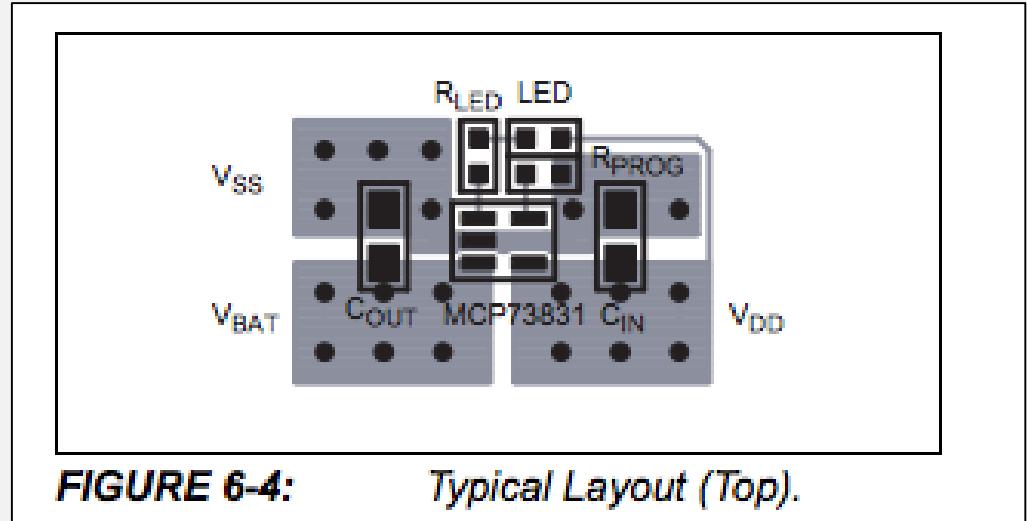


FIGURE 6-4: Typical Layout (Top).

FERRITE BEADS, ANALOG, DIGITAL

- For a mixed signal IC, it's good practice to use a ferrite bead to separate digital & analog voltage sources
 - Possible that digital noise could leak onto analog supply and interfere with readings
 - PCB design should also sequester digital signals

